

Publikace v prostředí webu s využitím .NET MVC 3

Web publication based on .NET MVC 3 technology

Zadání bakalářské práce

Student:

Vítězslav Šimon

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Publikace v prostředí webu s využitím .NET MVC 3
Web Publication Based on .NET MVC 3 Technology

Zásady pro vypracování:

Technologie webových aplikací je neustále se vyvíjející oblastí. Cílem práce je zachycení možností technologie .NET MVC 3 pro realizaci a provoz publikačního systému v prostředí internetu.

1. Popište architekturu a základní koncept technologie .NET MVC 3.
2. Analyzujte potřeby pro provoz publikačního systému pro malé a střední weby a na základě tohoto navrhnete funkcionalitu vlastního systému.
3. Realizujte vývojový proces navrženého systému s využitím platformy .NET MVC 3 a případně dalších vhodných technologií.
4. Popište a zhodnoťte výsledné řešení a jeho možnosti reálného nasazení.

Seznam doporučené odborné literatury:

- [1] Jon Galloway a kolektiv: Professional ASP.NET MVC 3, Wrox, ISBN: 978-1118076583, 2011
- [2] Andrew W. Troelsen: Pro C# 2010 and the .NET 4 Platform, Apress, ISBN: 978-1430225492, 2010

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Radecký**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 04. května 2012

.....*Vítězslav Šimon*.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 04. května 2012

.....*Vítězslav Šimon*.....

Chtěl bych na tomto místě poděkovat vedoucímu mé bakalářské práce,
panu Ing. Michalu Radeckému, Ph.D. za vedení mé práce poskytnuté rady a doporučení
pro její vypracování. Dále bych rád poděkoval své rodině za podporu při studiu.

Abstrakt

Tato práce se zabývá vývojem systému pro správu obsahu malých a středních webů. Vysvětluje pojmy „systém pro správu obsahu“, okolnosti vzniku těchto systémů, jejich typy a specifika těchto typů. Práce dále popisuje používané technologie pro vývoj těchto systémů a naznačuje směr dalšího vývoje v této oblasti. Práce definuje a rozebírá současné požadavky na systémy pro správu obsahu. Tyto jsou pak následně aplikovány v ukázkovém řešení. V poslední části práce je toto řešení analyzováno.

Klíčová slova: systém pro správu obsahu, publikační systém, elektronické publikování, WWW prezentace, ASP.NET, MVC

Abstract

This thesis deals with the development of systems for managing small and medium-sized sites. It explains the concepts of "content management system", circumstances of these systems, their types and specificities of these types. The work also describes the technology used to develop these systems and indicates the direction of further developments in this area. The work defines and analyzes the current requirements for content management systems. These requirements are subsequently applied in sample solution. In last part of thesis is this solution analysed.

Keywords: content management system, publishing system, electronic publishing, WWW presentation, ASP.NET, MVC

Seznam použitých zkratek a symbolů

ADO	– Access Data Object
API	– Application Programmable Interface
ASP	– Active Server Pages
CAS	– Code Access Security, Řízení přístupu ke zdrojovému kódu
CERN	– Evropská organizace pro jaderný výzkum
CMS	– Content Management System, Systém pro správu obsahu
CSS	– Cascading Style Sheets
DOM	– Document Object Model
EI	– Executing Item, spouštěcí položka. Jedná se pouze o termín v rozsahu této bakalářské práce, EI zastupuje v realizovaném systému SimplePublishCMS veškeré obsahové položky tohoto systému, jedná se o Článek, Skupina, Odkaz, Soubor ke stažení, RSS výstup. Ve skutečnosti se jedná o třídu, ze které dědí každý obsahový objekt systému
HTML	– Hyper Text Markup Language
MVC	– Model-View-Controller
ORM	– Object-Relational Mapping
RDF	– Resource Description Framework
RSS	– RDF Site Summary
SAX	– Sample API for XML
SQL	– Structured Query Language, strukturovaný dotazovací jazyk
URL	– Uniform Resource Locator
WWW	– World Wide Web
WYSIWYG	– What You See Is What You Get

Obsah

1	Úvod	9
2	Systém pro správu obsahu	11
2.1	Důvody vzniku CMS systémů	11
2.2	Co je CMS systém	11
2.3	Cíle CMS systémů	11
2.4	Požadavky na CMS systémy	13
2.5	Typy CMS systémů	14
2.6	Často používané CMS systémy	16
2.7	CMS systémy a finanční náklady na pořízení	16
2.8	Další směřování CMS systémů	18
3	Technologie CMS systémů	19
3.1	Perl	19
3.2	PHP	19
3.3	Java Server Pages (JSP)	20
3.4	Ruby	21
3.5	Python	21
3.6	Active Server Pages (ASP)	22
3.7	ASP.NET	22
3.8	ASP.NET MVC3	22
3.9	Další technologie spolupracující s ASP.NET MVC3	24
4	Systémy pro malé a střední weby	33
4.1	Požadavky na CMS systémy	33
4.2	Alternativy pro chybějící funkcionality	36
5	Realizace systému SimplePublishCMS	39
5.1	Vize realizace	39
5.2	Use-Case model	40
5.3	Technická specifikace	43
5.4	Bezpečnost systému	45
5.5	Schéma architektury	45
5.6	Další technologie použité v realizaci	45
5.7	Přínos technologií	48
5.8	Ukázky uživatelského prostředí systému	49
5.9	Administrace obsahu (strana přihlášeného uživatele)	49
6	Zhodnocení systému SimplePublishCMS	55
6.1	Problémy při realizaci systému	55
6.2	Možnosti dalšího rozvoje systému	55
7	Závěr	57

8 Reference	59
Přílohy	60
A Další technologie použité v realizaci	61
A.1 HTML	61
A.2 CSS	61
A.3 Microsoft SQL Server 2008	62
A.4 Internetová Informační Služba (IIS)	63
B Rysy používaných CMS systémů	65
C Ostatní	73
C.1 Obsah přiloženého CD	73
C.2 Další přílohy	73
C.3 Tiobe Index	73

Seznam tabulek

1	Seznam událostí a reakcí	42
2	Výčet technologií SimplePublishCMS	44
3	CMS - Systémové požadavky, Realizace bezpečnosti, Podpora	68
4	CMS - Snadnost použití, Výkon, Řízení	69
5	CMS - Podpora formátů, Flexibilita	70
6	CMS - Moduly v instalačním balíčku	71
7	CMS - Moduly v instalačním balíčku 2	72
8	Nejoblíbenějších 20 programovacích jazyků (TIOBE Index)	74

Seznam obrázků

1	Obecný blokový diagram datových toků v CMS systému	13
2	Schema komunikace Klient - Server	20
3	Blokový diagram architektury .NET MVC3	23
4	Blokový diagram architektury .NET Framework z pohledu verzí	25
5	Blokový diagram architektury .NET Framework z pohledu zpracování kódu	26
6	Blokový diagram architektury .NET Framework z pohledu aplikací a systému	29
7	Ukázka dědičnosti v EF a jejího řešení na úrovni SQL dotazu	31
8	Use-case diagram: aktivity: SuperUser a Návštěvník	41
9	Use-case diagram: aktivity: ContentContributor	41
10	Aktéři a dědičnost	42
11	Class diagram (zjednodušený)	43
12	Kontextový diagram	44
13	Schéma architektury	45
14	TinyMCE WYSIWYG HTML editor	47
15	Prostředí systému: Procházení obsahu	49
16	Prostředí systému: Pohled na EI (Článek)	50
17	Prostředí systému: Pohled na úplný výpis EI (Článek)	50
18	Prostředí systému: Pohled na EI (RSS výstup)	50
19	Prostředí systému: Výběrové pole pro datum a čas	51
20	Prostředí systému: Informační zpráva systému (položka bez lokalizace)	51
21	Prostředí systému: Administrace	52
22	Prostředí systému: Administrace spustitelných položek (EI)	52
23	Prostředí systému: Administrace dalších možností EI (článek)	53
24	Prostředí systému: Pohled na jeden z modulů (Menu)	53
25	Prostředí systému: Administrace : Akce Edit (položka menu)	54
26	Prostředí systému: Administrace : Akce Delete (číselník)	54
27	Administrace IIS	64
28	Administrace systému Drupal	66
29	Administrace systému PHP-Fusion	66
30	Hlavní strana systému PHP Nuke	67
31	Hlavní strana systému WordPress	67

Seznam výpisů zdrojového kódu

1	Python, ukázka syntaxe	21
2	Razor View Pages, ukázka syntaxe	28
3	Tag pro import JavaScript souboru do hlavičky HTML	46
4	Definice JavaScriptu v hlavičce HTML	46
5	JavaScript jako součást tagu	46
6	Ukázka kódu jQuery	47
7	CSS, ukázka syntaxe	61
8	Tag pro import CSS souboru do hlavičky HTML	62
9	Definice CSS v hlavičce HTML	62
10	Užití atributu style v HTML souboru	62

1 Úvod

Stále se rozvíjející dynamické prostředí Internetu oslovuje více a více lidí pomocí nejrůznějších portálů a služeb. Vzhledem k tomuto faktu je Internet mocným nástrojem, protože skýtá možnost rychle oslovit široké spektrum koncových uživatelů. Z tohoto důvodu je Internet velmi konkurenční a dravé prostředí.

Aby v takovémto prostředí bylo možno uspět, je potřeba tuto nevýhodu vyvážit. To znamená mimo jiné vytvářet kvalitní a efektní webové prezentace, které budou schopny oslovit takzvaně na první pohled. Pro tvorbu efektních a efektivních webových prezentací je potřeba výkonných softwarových nástrojů, které nám různými způsoby pomohou v dosažení tohoto cíle. Tyto softwarové nástroje bývají souhrnně označovány jako **systémy pro správu obsahu**. Právě analýzou a vývojem takovýchto systémů se zabývá tato bakalářská práce.

Práce se nejprve zaměřuje na samotný pojem **systém pro správu obsahu**, jeho vysvětlení a okolnosti jeho vzniku. Dále jsou zkoumány základní cíle těchto systémů a požadavky, které jsou na ně jejich uživateli často kladeny. Práce také uvádí dělení těchto systémů do skupin podle jejich zaměření a následně rozebírá jejich další specifické požadavky, rysy a možnosti, které jednotlivé skupiny nabízí. Také je zmíněna otázka licencí a poplatků v souvislosti s těmito systémy. Následně je zmíněno i budoucí směřování těchto systémů.

Ve druhé kapitole jsou rozebrány nejčastěji používané technologie pro vývoj jakýchkoli webových aplikací s tím, že se poté práce zaměřuje především na technologii **ASP.NET** a následně pak na její nadstavbovou technologii **ASP.NET MVC 3**.

Ve třetí kapitole jsou zmíněny realizace pro malé a střední weby především z hlediska častých požadavků na systémy pro správu obsahu.

V dalších kapitolách je představen ukázkový systém pro správu obsahu **SimplePublishCMS**. Následně jsou také popsány i další technologie, na kterých je systém postaven. Systém je zde popsán z hlediska analýz i důsledků plynoucích z užití vybraných technologií.

V závěrečných kapitolách je ukázkový systém zhodnocen z hlediska užitnosti a jsou zde nastíněny další možnosti směřování jeho vývoje.

2 Systém pro správu obsahu

V této kapitole je přiblížen pojem **systém pro správu obsahu (CMS)** a rozebrány základní vlastnosti a typy těchto systémů. Dále jsou popísány požadavky, které bývají na tyto systémy často kladeny. Závěrem kapitoly je naznačen budoucí vývoj těchto systémů.

2.1 Důvody vzniku CMS systémů

V minulosti, když ještě neexistovaly žádné CMS systémy, bylo nutné WWW stránky vytvářet ručně. Takto projektované stránky byly označovány jako tzv. **statické WWW stránky**. Toto řešení však nebylo možné udržet dlouhodobě, především z důvodu stále rostoucího množství dat, které bylo v minulosti limitováno tehdejší kapacitou úložišť a náklady na jejich pořízení.

Dalším nedostatkem bylo, že uchovávaná data často nedodržovala jednotnou formu vzhledu, což znamenalo při změnách náročnou správu, zálohování i horší orientaci v těchto datech.

Pravděpodobně nejzásadnějším impulsem pro změnu tohoto stavu však byla skutečnost, že bylo nutné při publikaci nové informace vytvořit HTML stránku, která danou informaci obsahovala a poté ji nahrát na souborový server.

Tyto důvody vytvořily prostor pro vznik CMS systémů.

2.2 Co je CMS systém

Definice 2.1 *CMS systém je možno definovat jako systém pro automatizaci správy a prezentace elektronického obsahu webu uživatelsky komfortní formou. Prezentace obsahu uživateli probíhá prostřednictvím dynamicky generovaných WWW stránek.*

Procesem správy elektronického obsahu je míněno především vytváření, aktualizace, publikování, vzájemné propojování, dále pak katalogizace a indexace elektronického obsahu.

Elektronickým obsahem, čili spravovanými daty je myšlen především text, dále také obrázky, video a audio soubory.

2.3 Cíle CMS systémů

Mezi základní cíle libovolných informačních systémů (a také CMS) patří [8]:

1. zajištění uživatelsky komfortní práce s evidovanými daty
2. zajištění oddělení informací od jejich vzhledu (pomocí šablon)
3. oddělení vnitřní implementace od výpisu dat uživateli (při výpisu dat se již neřeší na stejném místě přístup k databázi, a podobně)
4. umožnit rozšiřitelnost o funkcionality třetích stran (práce s novými typy objektů, jinými databázovými technologiemi, ...)

5. poskytnout prostředí pro snadné rozšíření o další funkcionality systému (moduly) bez nutnosti zásadním způsobem zasáhnout do jádra systému

2.3.1 Uživatelsky komfortní práce s evidovanými daty

Uživatelsky komfortní forma správy dat je obvykle řešena samozřejmými funkcionalitami, jako je výběrové pole pro cizí klíče, či datum, stránkování dlouhých výpisů nebo grafickým textovým editorem pro HTML kód. Uplatnění dalších vylepšení pak závisí na zaměření daného CMS systému. Typy CMS systémů podrobněji rozeberu v kapitole 2.5.

2.3.2 Prostředí pro snadné rozšíření o další funkcionality

Jedním z nejdůležitějších cílů publikačního systému je poskytnout prostředí pro snadné rozšíření o další funkcionality systému (moduly) bez nutnosti zásadním způsobem zasáhnout do jádra systému.

Vhodně naprogramované nové moduly mohou obecně zaměřenému systému „otevřít dveře“ k řešení specifických úloh, které nejsou tak frekventované, jako ty, pro které byl systém zamýšlen původně. Nebo naopak velmi specifickému systému mohou moduly napomoci, aby si udržel své postavení na trhu.

Bez nadsázky lze říci, že řešení tohoto cíle rozhoduje o budoucí úspěšnosti celého publikačního systému. Pokud bude tato část systému dobře naimplementována a autoři budou pružně reagovat na nastalé změny trendů ve vývoji webových portálů, má jejich systém velkou šanci se prosadit i v prostředí nových konceptů.

2.3.3 Oddělení informací od vzhledu

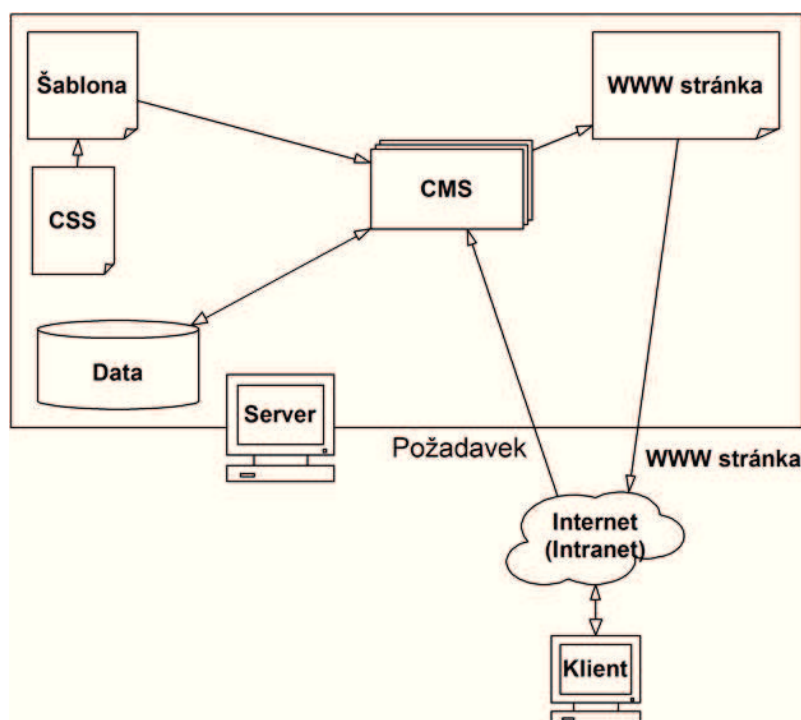
Oddělení informací od vzhledu jejich prezentace je nejčastěji zajišťováno pomocí šablon. Šablony jsou v podstatě HTML soubory doplněné o specifické konstrukce daného jazyka (cykly, podmínky, výpis proměnných a objektů reprezentujících data z databáze, vložení jiné šablony, speciální HTML tagy, ...). Při zpracování požadavku od uživatele jsou nejprve systémem vybrána data z databáze a následně jsou vložena do šablony náhradou za konstrukce uvedené v šabloně. Na obrázku 1 je celá věc blíže naznačena.

Hlavní výhodou takto koncipovaného řešení je možnost rychle a efektivně měnit vzhled výstupu bez nutnosti zasahovat hlouběji do systému. Další výhodou je možnost přizpůsobit výstup systému i pro jiná zařízení, například PDA, mobilní telefon a další. [8]

O konkrétní realizaci šablon budu dále hovořit v kapitole 3.9.2 o šablonovacím systému Razor, který jsem si za účelem řešení této části realizace své bakalářské práce zvolil.

2.3.4 Oddělení vnitřní implementace od procesu tvorby výstupu

Cílem tohoto bodu je dosáhnout stavu, že vývojář, který šablonu upravuje, nemusí znát, jak nebo odkud se data získala. Vývojáři stačí znát pouze to, jaká data má možnost v šabloně vypsat. Data pro šablonu mohla být získána následujícími způsoby:



Obrázek 1: Obecný blokový diagram datových toků v CMS systému

1. SQL dotaz (výběr z databáze)
2. Výpočet (SQL dotaz s funkcí COUNT(*) či jinou nebo výpočet provedený programově, ten však není tak rychlý, jako SQL dotaz)
3. Jiným programovým postupem (například sestavení kolekce z několika typů dat)

Z této množiny dat si vývojář, který šablonu upravuje, vybere jen ta, o která má zájem.

Další možností, kterou by vývojář mohl využít a kterou by měl systém poskytovat, je možnost vložit vlastní definované prvky. Ovšem daná technologie na to musí být stavěná, musí být vybavená interpretem, který toto bude schopen umožňovat.

2.4 Požadavky na CMS systémy

Požadavky na systémy jsou různé, především v závislosti na rozsáhlosti řešení, či podrobnosti řešení nebo podle maximálního předpokládaného rozsahu dat, která bude systém spravovat.

Mezi nejzákladnější funkcionality patří:

- fulltextové vyhledávání
- správa kategorií

- vizuální editor pro úpravu textu
- editace a správa různého obsahu
- správa uživatelských rolí a práv
- podpora pro RSS odběr informací

Tyto funkcionality by měly být implementovány v každém systému.

Dále je na zvážení, zdali by měl systém podporovat multijazykovost dat, pokud se ale vývojář rozhodne pro realizaci tohoto později, může to znamenat zásadní zásah do koncepce vyvíjeného systému, proto je vhodné s touto funkcionalitou počítat pokud možno již na začátku vývoje.

K rozboru těchto požadavků se ještě vrátím v kapitole 4.1.

V následující kapitole je obsaženo pojednání o typech publikačních systémů a rozboru jejich klíčových funkcionalit.

2.5 Typy CMS systémů

Jak již bylo řečeno, CMS systémy lze rozdělit podle zaměření do několika kategorií. Níže uvádím jejich výčet i s popisem klíčových funkcionalit:

Blog

Blog je odvozen zkrácením anglického Web Blog. Jedná se o jednoduchý „osobní deníček“ s mělkou skupinovou strukturou. Hlavním požadavkem na tento systém je velmi jednoduchá a efektivní práce především s textem. Další vítanou funkcionalitou je podpora alespoň jednoúrovňových diskusí. Vhodnou funkcionalitou může být i kalendář, který by sloužil pro rychlé vyhledání blogových zápisů. Jako příklad je možné uvést CMS WordPress.

Fotogalerie

Tento systém se zabývá řízením prezentací fotografií. Hlavní funkcionalitou je členění fotografií do kategorií (alb), podpora uploadu souborů přes webové rozhraní a zajištění podpůrných funkcí pro úpravu těchto fotografií (například: vytvoření zmenšenin daných obrázků, provedení ořezu obrázků, vložení vodoznaku do obrázku, převod do stínů šedi a podobné). Tato funkcionalita bývá nejpoužívanějšími CMS systémy zajišťována ve formě doinstalovatelného modulu.

Portál

Jedná se o web se svým katalogem odkazů, firem a dalšími podpůrnými funkcionalitami, jako jsou například: vyhledávání, email, zprávy, diskuse, ankety, TV program, mapy, slovník a další. Velice vhodnou funkcionalitou je SEO optimalizace a může být i kalendář akcí, který lze v systému mimo jiné používat i pro rychlé vyhledání záznamů. Tyto portály bývají často řešeny na míru vyvíjeným CMS systémem, který provozovatel

dále nešíří.

Diskusní fórum

Tento systém spravuje web, jehož prostřednictvím je možno vložit vlastní názor a reakci na specifická témata. Častěji bývají příspěvky koncipovány ve formě stromu, někdy v chronologické jednorázové podobě. Jako příklad je možné uvést PHP BB.

E-Shop

Tento systém slouží pro řízení elektronického obchodu. Jedná se především o sortiment, objednávky a sledování průběhu jejich vyřízení, řízení záznamů o reklamaci, zajištění možnosti elektronické peněžní transakce a další. Právě v souvislosti s peněžními transakcemi je zde kladen důraz na zvýšenou bezpečnost přenosu dat i přihlašovacích mechanismů. Dobrou a vhodnou inspirací v tomto směru může být web alfacomp.cz.

Encyklopedické systémy

Jsou to systémy pro tvorbu elektronických encyklopedií. Zde je kladen především důraz na zpracování textu, související odkazy, seznam referencí a citací, hypertextové odkazy v textu, u specifických slov, či termínů a také na vysvětlení daného termínu. Vhodnou funkcionalitou je také podpora exportu z tohoto systému. Nejvhodnějším příkladem je: WikiCMS.

E-learning

Jedná se o systém pro řízení výuky, výukových materiálů, kurzů, rozvrhů, studentů a jejich hodnocení. V této realizaci je kladen důraz především na systém rolí a oprávnění a také na bezpečnost přihlašovacího mechanismu. Jako příklad lze uvést EdISoN.

Systém pro sociální síť

Jedná se o systém, který primárně eviduje osobní informace o registrovaných jednotlivcích (nebo firmách, skupinách osob, organizacích). Tento systém je, nebo měl by být, náročný na bezpečnost, protože pracuje s osobními daty. Uchovávaná osobní data jsou pak využívána například při:

1. hledání osob, které by daná osoba mohla znát přes společnou vazbu v daném čase (například kolegové z práce, spolužáci ze školy a jiné). Při analýze známostí je často zohledněna i doba, po kterou se lidé neviděli (princip zapomínací funkce).
2. hledání osob, se kterými má daný člověk společné zájmy
3. aplikaci cílené reklamy na danou osobu (třeba i v kombinaci s uvedenými zájmy)

Tento CMS je poměrně náročný na množství aplikovaných funkcionalit, patří mezi ně například správa fotogalerií, diskuse, podpora práce s textem, kalendář akcí nebo také, jak již bylo zmíněno, subsystém pro řízení reklamních ploch. V souvislosti s těmito systémy je vhodné doplnit, že špičkové CMS systémy z této kategorie jsou vlastnictvím provozovatele a ten je zpravidla nedává k dispozici. Jako ideální příklady lze uvést

FaceBook, Google plus, Twitter. Volně dostupný a kvalitní systém v této oblasti je podle názorů na Internetu CMS Drupal.

2.6 Často používané CMS systémy

S ohledem na informace v předchozí kapitole lze často používané systémy poměrně dobře odvodit z představených typů. Nicméně velká část z existujících, často používaných CMS systémů, pokrývá většinu funkcionalit těchto typů. Pokud ne přímo, tak alespoň pomocí modulů. Tyto moduly jsou většinou volitelně ke stažení z příslušných oficiálních stránek systému.

Mezi často využívané CMS systémy patří:

1. Drupal
2. Joomla
3. PHP Fusion
4. WordPress
5. PHP Nuke

Charakteristiky různých nároků a rysů těchto systémů jsou k dispozici v příloze B.

2.7 CMS systémy a finanční náklady na pořízení

V souvislosti se systémy pro správu obsahu jsem doposud nezmínil otázku finančních nákladů na pořízení těchto systémů.

Trh CMS systémů, jako každý jiný, je rozdělen na dva segmenty - neplacené a placené systémy. Při pohledu na stav obou těchto segmentů lze tvrdit, že neexistuje žádné placené řešení, které by nemělo srovnatelnou alternativu mezi řešeními neplacenými. Výjimku mohou představovat pouze technologie související se SEO (optimalizace stránek pro vyhledávače) a reklamní management. V minulosti byly předmětem kritiky u některých open-source CMS také technologie bezpečnostního řešení. V tomto ohledu však lze vyjádřit myšlenku Paretova pravidla, které přeneseno do tohoto kontextu říká: "zlepšení 20 procent představuje 80 procent nárůst finančních nákladů".

Nyní rozeberu výhody a nevýhody obou segmentů.

Neplacené systémy jsou zaměřeny především na nejčastější požadavky. Bývají často navíc **open-source**. Open-source systémy můžete po nabytí základních znalostí technologie programovacího jazyka a architektury CMS systému rozšiřovat a upravovat podle svého.

Část open source systémů (například PHP Nuke) je dodávána s informací v patičce stránky, která říká přibližně: "Powered by ...". Tuto informaci smíte podle licence odstranit až ve chvíli, kdy zaplatíte vývojáři v rozmezí od 10 do 20 dolarů, záleží na systému. Nicméně i s touto drobnou vadou na kráse jsou systémy plnohodnotně použitelné ve

všech svých funkcích. Kolem open-source systémů je shromážděna velká komunita uživatelů na diskusních fórech, kteří vám v nesnázích při nasazování systému rádi a ochotně poradí. Taktéž jsou při těchto systémech shromážděny velké komunity vývojářů, kteří za svoji práci na systému často nedostávají odměnu a dělají ji ve volném čase. Nicméně kvalita těchto systémů při pohledu na ty nejpoužívanější není tímto vůbec dotčena.

Pro uživatele, kteří se v problematice práce s CMS systémem neorientují jsou k dispozici školení, v cenovém rozmezí od bezplatného až do 1000 Kč za kurz. Tato školení často svolává místní komunita uživatelů nebo někdo z vývojářů daného CMS.

Placené systémy jsou vhodné ve chvíli, kdy nevyhovuje konvenční řešení. Toto bývá nejčastěji ze dvou důvodů - určitého požadavku na nějakou funkcionalitu nebo toho, že klient nezná technologie pro realizaci webu a nechce se je učit. To slovy Paretova pravidla znamená, že jste mezi 20 procenty specifických případů. Výrobce těchto systémů vyvine klientovi systém přesně podle jeho požadavků a potřeb. Nicméně toto řešení je pochopitelně dražší. Dále vývojářské společnosti v rámci svých služeb nabídnou nasazení finálního systému na hosting, registraci vlastní domény, smlouvu s webhostingovou společností nebo svůj webhosting. Součástí služeb bývá školení zaměstnanců pro práci s daným systémem. Část služeb, jako například již zmiňovaný webhosting nebo technická podpora jsou placeny paušálně, často formou předplatného.

Nyní se ještě vrátím k cenám komerčních CMS systémů. K 28.04.2011 uváděla společnost NetDirect s. r. o. [19] na svých stránkách cenu realizace postavené na svém CMS MediaCentrik na 30000 Kč. Součástí této ceny je vše uvedené v předchozím odstavci. Navíc je součástí služeb i marketingová podpora pro klienta v následujících letech. MediaCentrik je postaven na technologiích ASP.NET a MS SQL 2008, tedy na podobných, jako praktická realizace v této bakalářské práci.

Dalším dodavatelem komerčních CMS systémů je společnost ALTOS Software [20], která na trhu funguje od roku 2001. Tato firma používá technologie od společnosti Microsoft, které blíže neupřesňuje, pouze odkazuje na to, že jsou nejnovější. Altos také vyzdvihuje používání technologie asynchronního JavaScriptu ve svých realizacích. Cena jejich služeb na webu nebyla uvedena, a to ani rámcově.

Posledním komerčním produktem, který zde uvedu je publikační systém WebGet [21] od společnosti IT STUDIO s.r.o.. Tento systém je prodáván paušálně s tím, že se předplácí na jeden, půl nebo čtvrt roku. Jeho cena se tak pohybuje měsíčně v průměru od 196 až 235 do 1042 až 1250 Kč (bez DPH). Ceny jsou uvedeny v ceníku vydaném 10.03.2011 a zahrnují aktualizace a webhosting. Aby tento systém postačoval běžným potřebám, je nutné si jej zaplatit ve verzi Standard a vyšší, která začíná na ceně 792 Kč měsíčně. K CMS WebGet je možno si přikoupit řadu dalších funkcí.

Na závěr kapitoly uvedu jen rychlé zhodnocení uvedených komerčních produktů. Jako nejvhodnějšího dodavatele systémů uvádím společnost NetDirect a to z důvodu velkého množství referencí i schopnosti odhadnout minimální cenu své práce. Druhým v pořadí je podle mě společnost ALTOS a to z důvodu menšího počtu referenčních klientů a toho, že minimální cenu neuvedla.

2.8 Další směřování CMS systémů

Další směřování vývoje bude bezpochyby ovlivněno technologií HTML 5 a novými možnostmi, které nabídne. HTML 5 zmiňuji v kapitole A.1. Dále se bude pravděpodobně jednat o aplikaci technologie AJAX (asynchronní JavaScript). Věřím, že důležitou roli při dalším vývoji CMS systémů bude hrát i snaha o jejich přizpůsobení pro mobilní zařízení, protože jejich počet rychle roste každým rokem.

Bezpochyby bude vznikat řada nových nebo aktualizace stávajících modulů pro CMS systémy.

Věřím také, že do budoucna vývoj zacílí i na rozhraní pro vzájemné propojení více různých instancí systémů na Internetu. U několika CMS systémů jsou tyto funkcionality již implementovány, výhledově to však bude samozřejmé, podobně jako správa obsahu metodou Drag & Drop.

3 Technologie CMS systémů

V této kapitole uvedu postupně technologie nejvíce používané pro vývoj CMS systémů v současné době. Případně zmíním několik významných předchůdců, kterými se inspirovaly. Tyto technologie spojuje jeden zásadní rys - jedná se o technologie na straně serveru, se kterým pak komunikují klienti. Tato komunikace probíhá tak, že klient vyšle požadavek na server, ten jej předá ke zpracování interpretu technologie a ve výsledku jako odpověď vrátí vygenerovanou HTML stránku s nějakým obsahem, která je pak uživateli zobrazena ve webovém prohlížeči. Schéma můžete vidět na obrázku 2.

Nejvýznamnější technologie obecně představím v následujících kapitolách.

3.1 Perl

Perl je interpretovaný, multiparadigmatický programovací jazyk. Tento jazyk vytvořil Larry Wall v roce 1987. V začátcích vývoje se jednalo o skriptovací jazyk. Představoval náhradu jazyka AWK a interpretru sh. Hlavním mottem jazyka je "vše lze udělat více než jednou cestou". [24]

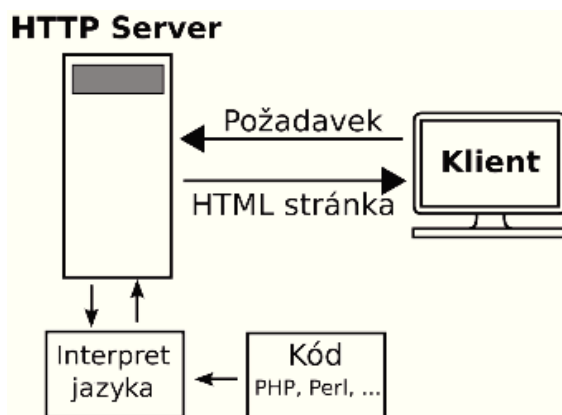
Komunita programátorů rostla nejvíce v roce 1991 (verze 4). Ve verzi 5 byl jazyk obohacen o výkonné datové struktury a objektové programování. Poměrně rozšířený je Perl v bioinformatice. S rozvojem internetu Perl začal být populární pro tvorbu CGI skriptů.

Hlavní výhoda tohoto jazyka plyne především z toho, že je interpretovaný, proto není potřeba kód kompilovat a linkovat, což zrychluje vývoj. Dalšími výhodami jsou pak automatická správa paměti, snadné propojování modulů, platformní nezávislost, snadná práce s textem, dynamické volání procedur a možnost použití interpretu například v jazyce C. Perl má také za sebou bohatou knihovnu kódů (tisíce volně dostupných modulů)[25], rozsáhlou dokumentaci i vývojářskou komunitu. Hlavní nevýhodou tohoto jazyka jsou vyšší nároky na paměť. K 26.4.2012 je Perl podle **TIOBE Indexu** [18] 10. nejoblíbenějším programovacím jazykem.

3.2 PHP

PHP, dříve označovaný jako Personal Home Page, dnes jako PHP: Hypertext Preprocessor, vzniklo tak, že Rasmus Lerdorf v roce 1995 vytvořil jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech ke svému webu, ke které postupem času připojil různé další funkcionality, například komunikace s databází a podpora pro dynamiku webu a kód následně uvolnil. V roce 1997 byla uvolněna verze 3.0, která zlepšila prostředí pro rozšiřování, sliněji definovala syntaxi jazyka a přidala se objektově orientovaná syntaxe. V roce 2000 byla uvolněná verze 4.0, která byla postavena na novém engine **Zend Framework**, který byl uveden o rok dříve. Verze 4.0 přinesla posun v modularitě kódové báze a vylepšení výkonu pro složité aplikace. V roce 2004 byla vydána verze 5.0, která vylepšila implementaci objektově orientovaného programování a byla změněna rozhraní pro práci s databázemi. [28] [29]

PHP je skriptovací, dynamicky typovaný programovací jazyk určený především pro programování dynamických webových stránek a aplikací postavených na (X)HTML.[17]



Obrázek 2: Schema komunikace Klient - Server

[28] PHP umožňuje také vývoj webových služeb. [28] [29] PHP je syntakticky podobné Perlu, Javě a C. PHP je s výjimkou několika specifických funkcí platformově nezávislý.

Pole v PHP jsou asociativní a jsou uchovávána ve formě hash tabulek. Dále PHP podporuje reference, dynamické volání procedur a dynamické načítání modulů. PHP má také podobně jako Perl rozsáhlou knihovnu kódů (jak interní, tak třetích stran) i komunitu vývojářů.

K 30.1.2011 byl PHP nejrozšířenějším jazykem pro programování webu (ostatně velká část CMS systémů je realizována právě v tomto jazyce), k 26.4.2012 je podle **TIOBE Indexu** [18] 6. nejoblíbenějším programovacím jazykem.

3.3 Java Server Pages (JSP)

Java je objektově orientovaný jazyk vyvinutý společností Sun Microsystems (dnes vlastněna a integrována v Oracle Software). Java je navržena s cílem, aby spouštění vyvinutých aplikací bylo nezávislé na platformě. Zdrojový kód je proto transformován do binárního kódu, který se na jednotlivých platformách spouští v prostředí označovaném jako virtuální stroj. Java podporuje běh ve vláknech a distribuovanost aplikací. [23]

Mottem Javy je: "Vývojáři se učí a používají Javu, aby byl Internet a zařízení lidem k užítku". [23]

JavaServer Pages je technologie vyvinutá společností Sun Microsystems v roce 1999. Tato technologie je srovnatelná s ASP i PHP, rozdíl je pouze v tom, že je při vývoji aplikací použita Java, pro kterou je JSP nadstavbovou technologií. Technologie se specializuje, stejně jako ty předchozí, na dynamické generování web stránek (HTML, XML, ...).

K 26.4.2012 je Java podle **TIOBE Indexu** [18] 2. nejoblíbenějším programovacím jazykem.

3.4 Ruby

Ruby je poměrně mladý interpretovaný, skriptovací, plně objektově orientovaný, dynamicky typovaný jazyk nezávislý na platformě. [26] Uveden byl Yukihirem Matsumotem v druhé polovině devadesátých let. Plně objektový znamená, že vše v kódu je objekt. Důraz je kladen na jednoduchost a přehlednost syntaxe. Technologie je snadná k naučení, je tedy vhodná i pro laiky. Z důvodu dřívější absence anglických dokumentací se jazyk rozšířil až později po svém zveřejnění.

Velmi oblíbený je ve spojení s frameworkem Ruby on Rails postaveným na softwarové architektuře Model-View-Controller, který slouží k dynamickému generování web stránek. [27]

Ruby je vhodný k vytvoření libovolně velkých projektů, CGI skriptů a ve spojení s knihovnamy jako například GTK, TK nebo Qt je vhodný i pro tvorbu GUI aplikací.

Dalšími výhodami Ruby jsou spuštění v interaktivním režimu, modularita zdrojů, či podpora regulárních výrazů.

Mezi hlavní nevýhody patří nižší rozšířenost v České republice, pravděpodobně z důvodu absence dokumentace v češtině.

K 26.4.2012 je Ruby podle **TIOBE Indexu** [18] 11. nejoblíbenějším programovacím jazykem.

3.5 Python

Python je plně dynamický, interpretovaný, multiparadigmatický, skriptovací, objektově orientovaný jazyk s automatizovaně spravovanou pamětí. [30] Syntaxe jazyka Python se může zdát zpočátku trochu netypická z důvodu požadavku, aby byl například blok kódu uvozen dvojtečkou a jeho obsah odsazen tabulátorem, či mezerou o 1 úroveň. Tento požadavek má ovšem pozitivní dopad na přehlednost zdrojového kódu. Ve výpisu 1 uvádím pro názornost ukázkou.

```
def RanToAndPrint(a) :  
    for i in range(a) :  
        print "Number" , i  
        print "Square" , i **2  
RanToAndPrint(7);
```

Výpis 1: Python, ukáзка syntaxe

Python je lehce zvládnutelný, dokonce do té míry, že je doporučován jako jeden z nejvhodnějších jazyků pro programátorské začátečníky. Další výhodou je vysoká produktivita vývoje, která je téměř nezávislá na rozsahu aplikace. Python je schopný spolupracovat s jinými programovacími jazyky (například C, C++, Java, .NET kompatibilní) a jejich aplikacemi, tak, že slouží jako jejich skriptovací jazyk, čímž zejména v kompilovaných aplikacích kompenzuje jejich někdy nedostatečnou pružnost.

Python sám umožňuje vývoj internetových serverů i klientů. Nicméně oblíbenou nadstavbovou technologií pro vývoj CMS systémů v Pythonu je například Django Framework. [31] Na tomto Frameworku je postaven i jeden z CMS systémů - Django CMS,

který může sloužit kromě aplikace CMS i jako demonstrativní příklad možností tohoto frameworku.

K 26.4.2012 je Python podle **TIOBE Indexu** [18] 8. nejoblíbenějším programovacím jazykem.

3.6 Active Server Pages (ASP)

ASP bylo současníkem a konkurentem PHP od Microsoftu. ASP bylo PHP podobné svou architekturou. Jedná se o serverový programovací jazyk. [4] [3] Oproti PHP nebyl Open Source, proto byla jeho podpora ze strany poskytovatelů webhostingu oproti PHP slabší. Od ASP se v poslední době již upouští, zejména proto, že se jedná o technologicky překonaný koncept. Nástupcem ASP je ASP.NET, které je ovšem výrazně odlišné oproti ASP.

3.7 ASP.NET

ASP.NET je součástí .NET Frameworku, zaměřenou na tvorbu webových aplikací a služeb. Aplikace založené na ASP.NET jsou rychlejší, neboť jsou předkompilovány do jednoho či několika málo DLL souborů oproti skriptovacím jazykům, kde jsou stránky při každém přístupu znovu parsovány (klasické ASP). Kompilace také pomáhá zachytit část chyb již při vývoji. [4] [3]

ASP.NET WebForms byl založen z důvodu snahy Microsoftu posunout vývoj webu dopředu. Hlavní roli sehrály tehdejší požadavky z praxe. Cílem bylo nabídnout flexibilitu a robustnost. WebForms je koncepčně podobný programování obvyklému u desktop aplikací, to znamená, že u něj platí stejné zvyklosti, například ovládací prvky, událostní model a další. Za účelem zajištění této podobnosti vyrobil Microsoft vlastní knihovnu prvků. Dalšími výhodami jsou například jednodušší vývoj rozsáhlých aplikací, či datově řízených aplikací.

Mezi největší nevýhody WebForms patří především neúplné oddělení prezentační vrstvy od logiky (code-behind soubory), dále je těžší provádět automatizované testy těchto aplikací.

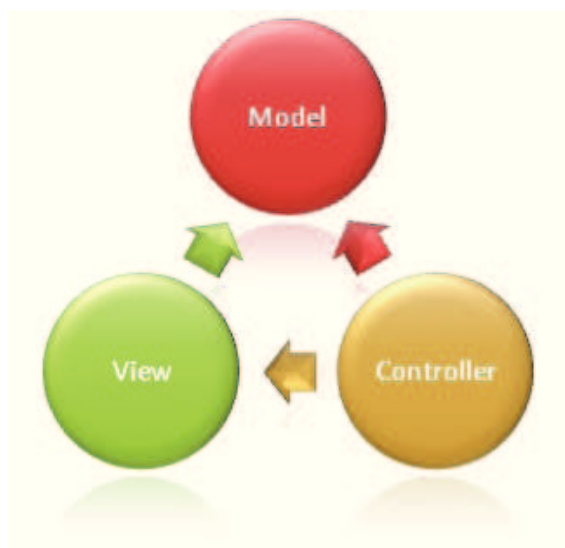
Poslední nevýhodou, kterou uvedu je způsob realizace bezstavovosti. Bezstavovost je řešena uložením v proměnné ViewState, ta je potom na klentské straně reprezentována jako skryté pole formuláře, čímž je navyšována velikost vygenerované stránky.

Druhým přístupem je koncept ASP.NET MVC, který jsem si zvolil pro realizaci své bakalářské práce. To, proč jsem tak učinil, popíši v následující kapitole.

3.8 ASP.NET MVC3

Model-View-Controller (MVC) je návrhový vzor architektury [2] [4], který rozděluje aplikační logiku do 3 prvků [10] [3] (viz. obrázek 3):

Model je objekt reprezentující datovou strukturu, se kterou aplikace pracuje. Tento objekt umí získat data z databáze, provádět nad nimi změny a uložit je zpět do databáze (obsahuje metody pro provádění těchto operací).



Obrázek 3: Blokový diagram architektury .NET MVC3

View je objekt upravující vzhled uživatelského rozhraní dané aplikace v souvislosti s daným modelem. S ohledem na typ zamýšlené aplikace se může jednat o formulář, tiskovou sestavu nebo v případě této práce o šablonu pro HTML stránku (vzhled HTML stránek je navíc upraven CSS styly, které ještě více omezují nutnost změny jejich zdrojového kódu). View je jako jediná část NET.MVC interpretovaná až v okamžiku zpracování požadavku.

Controller je objekt, který vymezuje množinu instancí modelu, které vyhovují danému požadavku. Controller obsahuje aplikační logiku v souvislosti s akcemi, které mohou být uživatelem vyvolány z View (včetně obnovy View, je-li to potřeba). Controller také určuje, který View bude použit k výpisu dat nebo interakci s uživatelem. Controller se také stará o volání všech funkcí modelu.

Vymezit neboli vybrat množinu instancí modelu je možné několika způsoby [1]:

1. SQL dotazem
2. funkcemi Objektově Relačního Mapování (ORM)
3. programově vytvořenou kolekcí objektů

3.8.1 Výhody MVC obecně

MVC jako takové má vzhledem ke své koncepci má výhodu v tom, že odděluje logiku práce s daty (Model) od činnosti aplikace (Controller) a od formy prezentace dat uživateli (View) [2]. V souvislosti s tímto je možné říci, že změny jedné z těchto částí se promítnou do ostatních dvou minimálně nebo v některých případech vůbec (například změny ve View).

V případě testování nebo opravy chyby je pak možno se zaměřit jen na specifickou část aplikace a provádět změny pouze v ní, nikoli v celé aplikaci.

Vzhledem ke své struktuře podporuje rychlý vývoj aplikací. Jednotliví programátoři se každý mohou zaměřit na jeden model a veškerou logiku s ním spojenou od řízení po zobrazení. Poté pomocí drobných změn ve View mohou funkcionality své části aplikace pohodlně propojit s ostatními částmi kolegů.

3.8.2 Výhody MVC3

ASP.NET MVC 3 Framework má kromě obecných výhod architektury MVC i další výhody [1] [3]:

1. Jedná se o jednotné dynamicky se rozvíjející prostředí s jedním velmi dominantním vývojářem – Microsoft (oproti například Java Server Pages) [4] [13].
2. Poskytuje událostní model, který zapouzdřuje komunikaci s protokoly (HTTP, FTP, SMTP a dalšími)
3. Poskytuje několik druhů řešení pro různé části aplikace (přihlašování, objektově relační rámce pro přístup k datům, zobrazování dat). Kromě dodaných řešení je možno napsat si také řešení vlastní.
4. Nativně pracuje s takzvanými „hezkými“ URL adresami, které jsou pohodlně zapamatovatelné pro uživatele
5. Umožňuje řízení výstupu (prezentační vrstva), je tak dosaženo menších a validních stránek
6. Nepřevzal rysy WebForms, jako ViewState neboPostBack
7. Podporuje prvky ASP.NET jako cachování nebo autentifikační mechanismy
8. Architektura těchto aplikací umožňuje unit testování přímo na aplikační vrstvě

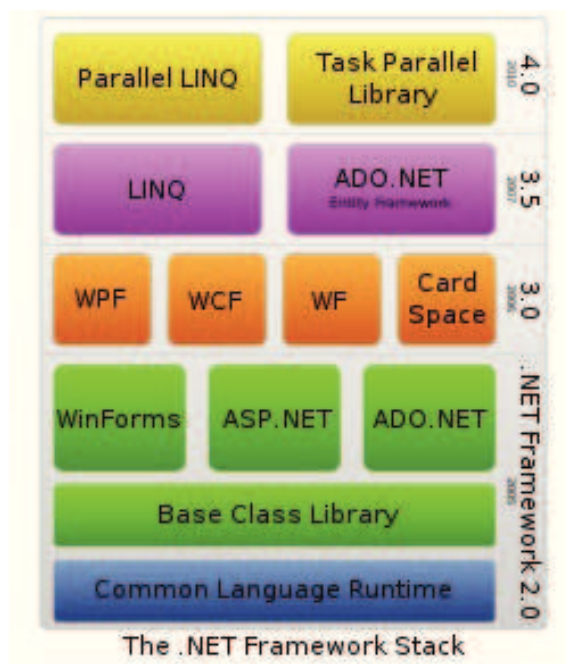
Jako nevýhoda se může jevit větší požadavek na znalost technologií, jako jsou HTML, CSS a JavaScript. Déle je v MVC 3 slabší podpora některých komerčních i open-source knihoven.

3.9 Další technologie spolupracující s ASP.NET MVC3

3.9.1 .NET Framework

Jak již bylo řečeno, .NET MVC3 je rozšířením .NET Frameworku. Aby byl popis úplný, je nutné zde popsat i tuto technologii.

Definice 3.1 (Microsoft) .NET Framework lze specifikovat jako softwarové prostředí pro vytvoření a běh aplikací nové generace. .NET Framework je založen na souboru myšlenek a pohledů na budoucnost IT a vývoje software, které se v Microsoftu prosadily počátkem roku 2000.



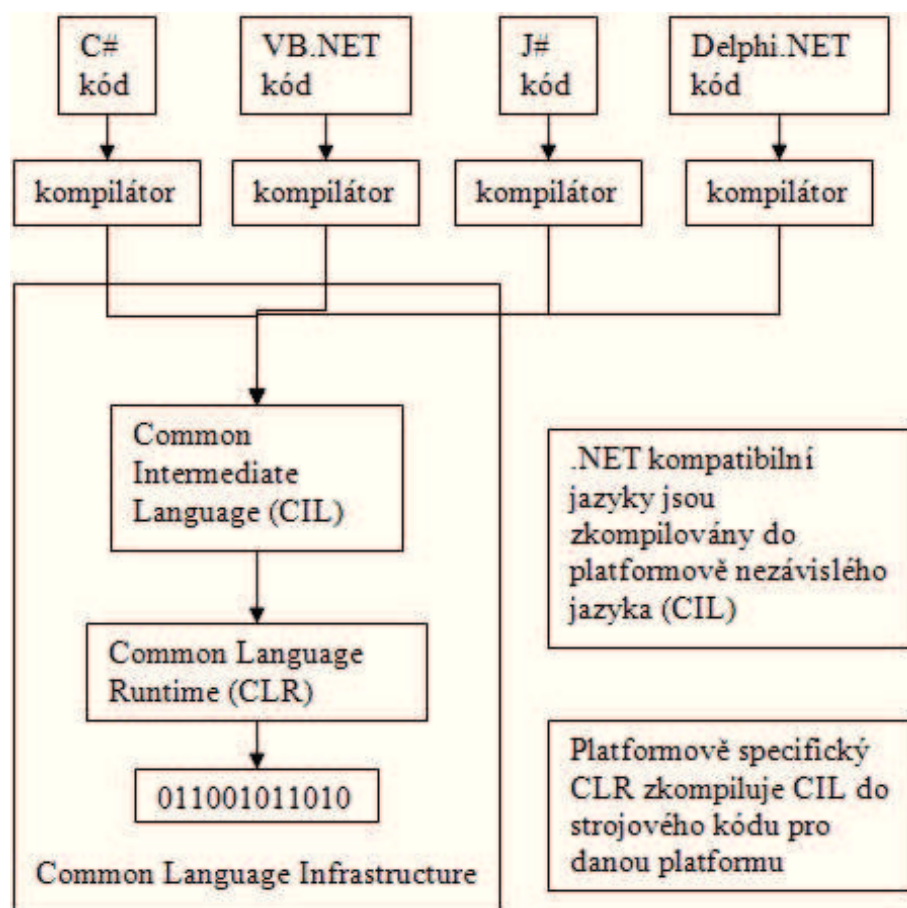
Obrázek 4: Blokový diagram architektury .NET Framework z pohledu verzí

[9] [6] Tyto myšlenky se prosadily částečně také v reakci na nové přístupy prezentované v Javě, ze které, kromě jiných .NET Framework čerpal řadu nápadů, které poté převzal do své implementace.

Hlavními cíli .NET Frameworku jsou snaha o vytvoření konzistentního objektově orientovaného prostředí pro vývoj, kde bude možné místně spouštět jak místní, tak i vzdáleně umístěný kód a sjednocení (co největší částí) procesu vývoje aplikací napříč typem (například: Windows aplikace, webová služba, ...) a technologií (COM, C++, průmyslová řešení, mobilní technologie, ...). [3] [14]

V návaznosti na prostředí pro vývoj má .NET Framework běhové prostředí, které zajišťuje:

1. Minimalizaci procesu vývoje SW
2. Minimalizaci konfliktu verzí součástí a SW - používáno centrální úložiště sestavení - Global Assembly Cache (GAC)
3. Možnost bezpečného spuštění neznámého kódu (kód třetích stran) a omezení práv přístupu k datům a jiným zdrojům
4. Omezení problémů s výkonem u rozhraní externích technologií



Obrázek 5: Blokový diagram architektury .NET Framework z pohledu zpracování kódu

Architektura .NET Frameworku - Common Language Infrastructure (CLI) je složena z několika částí. [32] Pro přehlednost je uvedena na obrázku 5.

První z nich je **Common Intermediate Language (CIL)**. Jedná se o jazyk společný celému .NET Frameworku. Do tohoto jazyka jsou překládány zdrojové kódy ve všech .NET kompatibilních jazycích [32] (například C#, Visual Basic, C++, Delphi a další). Tento překlad musí zajistit dodávané kompilátory, které koncepčně musí vycházet z ECMA standardu CLI.

Dalším prvkem, který uvedu je **Common Language Runtime (CLR)**, jedná se o službu, zajišťující základní servisní činnosti pro aplikace pro .NET Framework, mezi které patří například:

- Řízení kódu v okamžiku jeho sestavení a běhu
- Řízení paměti, vláken, vzdálených volání
- Zajištění bezpečnosti typů (definován společný systém základních typů), bezpečnosti a robustnosti běžící aplikace (proto je kód běžící pod CLR označován jako manage-ovaný)
- Zajištění podpory pro spolupráci s externími technologiemi (například: COM, DLL, ...)
- Kompilace Just-In-Time (JIT) zajistí, že manageovaný kód je překládán a běží v nativním strojovém jazyce daného systému. Překlad probíhá až při požadavku na spuštění kódu, případně před instalací nebo před prvním voláním dané metody
- Nezávislost na programovacím jazyku je nahrazena závislostí na .NET platformě
- Optimalizace práce aplikací se zdroji v systému - Garbage Collector zajišťuje uvolňování paměti s nepotřebným obsahem
- Pomocí CAS funkcionality zajišťuje také nastavení práv pro spuštění bloku kódu a to pomocí atributů bloku kódu. Na výběr jsou nejrůznější omezení (například: jméno uživatele, nutnost spuštění v UAC režimu, právo zápisu do souboru)

Posledním prvkem, o kterém se v souvislosti s .NET Frameworkem zmíním, je **Class Library**. Tato knihovna obsahuje objektové typy, které lze libovolně užívat v .NET aplikacích. Knihovna pokrývá například i tyto oblasti:

1. Práce se soubory (i specifickými, například: XML), přístup k souborovému systému
2. Práce s datovými proudy (streamy)
3. Síťová komunikace, práce s emaily
4. Šifrování a komprese dat
5. Aplikace běžně používaných návrhových vzorů (například: Observer)

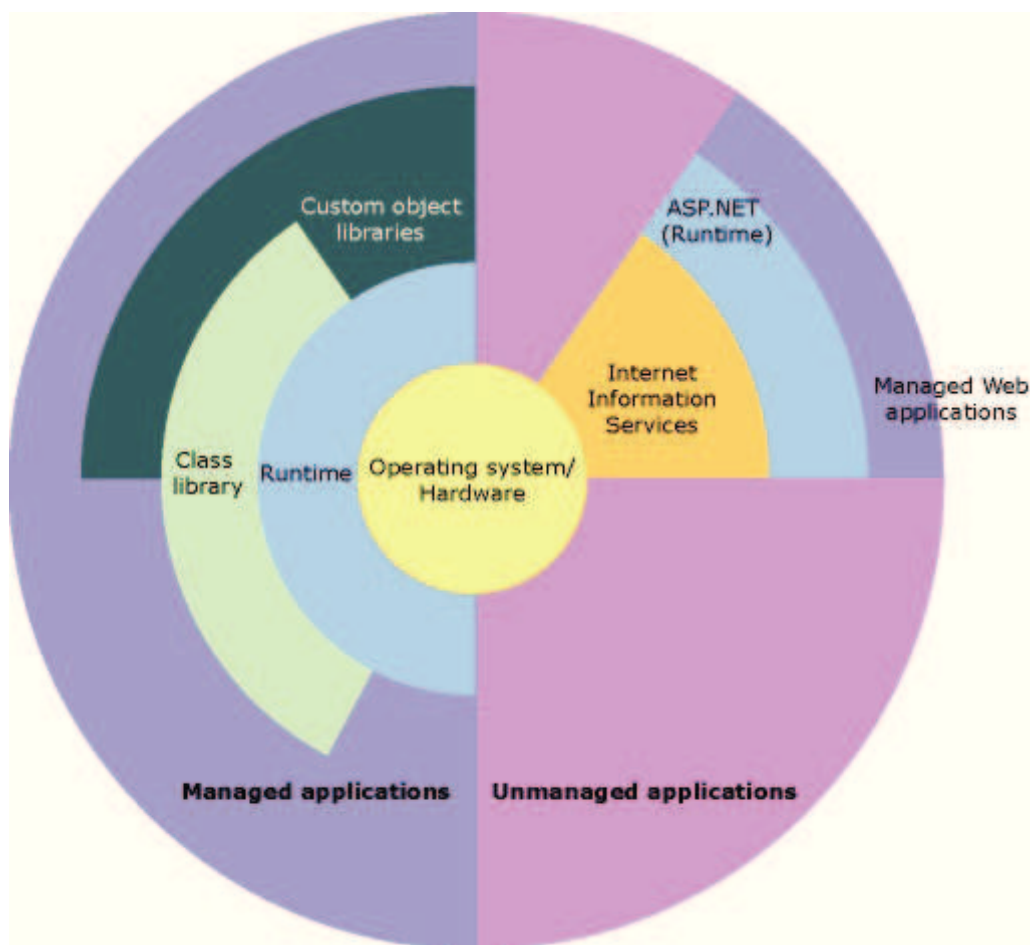
6. Konzolové aplikace (aplikace příkazového řádku)
7. Windows Forms aplikace, vykreslování a práce s grafickým uživatelským rozhraním a jeho prvky
8. webové aplikace a služby
9. Služby Windows
10. Práce s daty a komunikace s databázemi (MS SQL Server podpora integrována, ostatní je nutné doinstalovat), Linq
11. Reflexe a dynamické načítání knihoven
12. (De)Serializace instancí objektů
13. Dynamická tvorba kódu, či assembly za běhu
14. Podpora pro programové ladění přímo v kódu dané aplikace, logování ladících informací a další

3.9.2 Razor View Pages

Razor View Pages byl uveden v .NET MVC 3 jako alternativa k ASPX stránkám, která si klade za cíl zjednodušit oblast tvorby šablon pro výpis dat. [1] [3] Jedná se o propojení HTML kódu a jazyků C# nebo Visual Basic.NET. Toto propojení spočívá v možnosti psát do HTML kódu v takto koncipovaných samostatných blocích `@{ }` kód daného programovacího jazyka. [12] [11] Může se jednat o funkce, výpis proměnných, nebo vypsání HTML kódu a proměnných v cyklu, použití podmínek při výpisu, užití pomocných vestavěných (tzv. helpery) nebo vlastních funkcí. Stránky je také možné do sebe zanořovat. Kód stránek nemusí být jen celistvý s občasným vložením nějakého výpisu, ale je možné jej rozdělit do sekcí a ty poté vypisovat na konci View. Kód tak vypadá přehledněji a je tak lépe vidět struktura daného View. Při výpisu View je také možno měnit Layout View (hlavní View), který obsahuje základní rozložení daného webu (vhodné například pro výpis obsahu RSS kanálu). Razor se snaží o minimalizaci nutnosti přechodu mezi bloky Razoru a HTML kódu.

Příklad syntaxe můžete vidět ve výpisu 2. Razor View Pages v mé realizaci spolupracuje s Entity Frameworkem, od kterého získává data pro výpis v šablonách. Tato data mu zprostředkovává kontroler.

```
@model IEnumerable<SimplePublishCMS.Models.Language>
@{ ViewBag.Title = "Languages"; }
@functions{
    public SimplePublishCMS.Helpers.GlobalSPCMSHelpersNS GHNS = new SimplePublishCMS.
        Helpers.GlobalSPCMSHelpersNS();
    public MvcHtmlString WriteStdActionLinks(SimplePublishCMS.Models.Language item)
    {
        string AdminControllerName = "";
```

Obrázek 6: Blokový diagram architektury .NET Framework z pohledu aplikací a systému

```

        bool CanPrintImages = GHNS.GetCanActionsByImagesState();
        System.Text.StringBuilder TMP = new System.Text.StringBuilder("");
        TMP.Append(GHNS.WriteActionLink(new string[] { "Edit", "Delete" }, new string[] { "Edit", "Delete" }, new string[] { "Edit", "Delete" }, AdminControllerName, item.Id, ref CanPrintImages));
        return new MvcHtmlString(TMP.ToString());
    }
}
<h2>@ViewBag.Title</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
@{
    GHNS.SetStandardData(Html, Request, Session);
    var grid = new WebGrid(Model, canPage: true, rowsPerPage: GHNS.GetPageCapacity_OnPage());
    grid.Pager(WebGridPagerModes.NextPrevious);
    @grid.GetHtml(tableStyle: "",
        headerStyle: "",
        alternatingRowStyle: "",
        columns: grid.Columns(
            grid.Column("Name", "Language", canSort: true, format: @<b>@item.Name</b>),
            grid.Column("CultureName", "Short_name", canSort: true, format: @<b>@item.CultureName</b>),
            grid.Column("Actions", "Actions", canSort: false, format:
                @<text>@WriteStdActionLinks(item.Value)</text>)
        )
    );
}

```

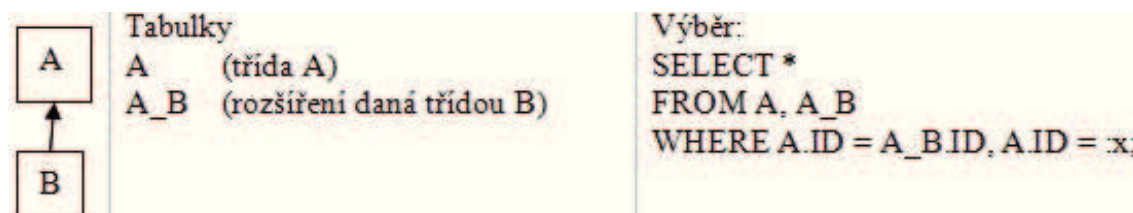
Výpis 2: Razor View Pages, ukázka syntaxe

3.9.3 Entity Framework

Entity Framework je dynamický objektově relační rámec. [3] [1] Cílem objektově relačního rámce je zajistit práci s daty (výběr, editace, přidání, odstranění, výpočty a sumarizace) v databázi - někdy specifické, často však řeší databázi libovolného typu (souborová, relační, objektová) nebo výrobce (Oracle, Microsoft, SyBase, IBM, SAP, MySQL AB, a další). Takto zajištěná práce s daty probíhá na úrovni objektů.

Objektově relační rámec postupuje následovně. Na pozadí vygeneruje normě odpovídající SQL dotaz pro provedení požadované operace nad stanoveným záznamem zastoupeným instancí třídy určitého typu, která v programu záznam dané tabulky reprezentuje, v případě výběru záznamu vloží hodnoty atributů tohoto záznamu do atributů instance třídy, tato instance je poté vrácena, obvykle v kolekci o jednom prvku. Pokud se požaduje vybrání množiny záznamů, tabulky nebo skupiny tabulek (děje se takto zpravidla podle kritéria), pak je namísto jedné instance třídy vrácena celá kolekce tříd obsahujících data získaná z databáze.

Vrácená třída může obsahovat všechna data záznamu, část těchto dat nebo několik dat z různých tabulek, která mají někde společnou vazbu (obvykle přes cizí nebo primární



Obrázek 7: Ukázka dědičnosti v EF a jejího řešení na úrovni SQL dotazu

klíč). Třída dále obsahuje například referenci na objekt reprezentující hodnotu cizího klíče (tzv. mapování vazeb). ORM také zajišťuje kromě mapování typů a vazeb navíc mapování tříd a jejich dědičností.

Nevýhody mají Entity Framework a ORM ve vínku díky své dynamičnosti nebo univerzálnosti (ve smyslu technologie databáze, nad kterou fungují). Hlavní nevýhodou je skutečnost, že omezují výkon databáze a zpomalují tak proces získávání dat a tím i odpovědi klientům. Aby se však jejich nevýhoda zmírnila, často jsou v nich používány následující techniky [13]:

1. **Connection pooling** – při inicializaci ORM je vytvořena sada předpřipravených spojení s databázovým serverem, omezí se takto doba nutná pro připojení k DB serveru, protože tato připojení jsou vracena připravenou funkcí, žádající o připojení k DB a při uvolnění jsou spojení vracena zpět do kolekce.
2. **Lazy load** – z databáze je vybíráno jen to, co je v dané chvíli nezbytně potřeba záznam a jeho hodnoty, hodnota cizího klíče je vyhledána jen, když je potřeba. Dále může být vyhledána jen 1 stránka předpokládaného výsledku či hodnoty cizích klíčů jen do určité hloubky.
3. **Operace UPDATE** je zacílena jen na opravdu změněné hodnoty, takže nedojde ke ztrátě aktualizace jinou aktualizací
4. Využívání explicitního uzamykání
5. V některých realizacích ORM **řídí úrovně izolace** kompromisně mezi spolehlivostí a rychlostí

Naopak výhodou Entity Frameworku je, že umožňuje ve své realizaci navrhnout konceptuální schéma (označeno jako EDMX diagram), které je automaticky převedeno do podoby databázové struktury včetně relací. Samozřejmě je také možný opačný postup. [1]

EDMX schéma uživatel tvoří v grafickém režimu, na pozadí jsou změny automatizovaně převáděny do zdrojového kódu jazyka C# nebo VB.NET, podle toho, který aplikace používá. Nevýhodou je, že jakákoli ručně provedená změna, či doplnění zdrojového kódu konceptuálního schématu je (nebo může být) po následném uložení z editoru ztracena.

Entity ze schématu jsou pak v databázi reprezentovány samostatnými tabulkami, přičemž pokud některá entita dědí z jiné entity, tak jsou v tabulce této entity zastoupeny pouze nové atributy a primární klíč. Dědičnosti ze schématu jsou pak řešeny dotazem nad tabulkami, které zastupují všechny entity v dědičnostní hierarchii. Tyto tabulky jsou dotazem sloučeny přes shodu v primárním klíči a výsledek je dále zpracováván. Bližší představu získáte pohledem na obrázek 7.

3.9.4 Linq

Language Integrated Query je integrovaný dotazovací jazyk, který byl uveden poprvé v .NET Framework 3.5. [3] [1] Linq byl od začátku svého vývoje zamýšlen jako nástroj pro práci nad jakýmkoli daty a zjednodušuje veškeré úkony nad nimi (výběry, třídění, tvorbu, rušení, propojení). Z tohoto důvodu jej také používám ve své realizaci. [13]

Linq je dostupný především pro tyto typy zdrojů [13] [1]:

1. Linq to XML – pracuje s daty v XML souborech bez využití standardních technik práce s XML soubory mezi které patří SAX (Sample API for XML) nebo DOM (Document Object Model), jedná se tedy o novou plnohodnotnou alternativu ve věci přístupu k XML datům.
2. Linq to Objects (Linq to DataSet) – práce nad daty v paměti Objects bývají kolekce nebo pole, DataSet kolekce dat získaných z databáze (technologie ADO.NET). Objekty implementují rozhraní: IEnumerable. Tento způsob je použit i v mé realizaci.
3. Linq to SQL – provádí dotazování nad databázemi, které používají rozhraní MS SQL Server. Linq příkazy jsou poté převedeny na SQL příkaz, který je zpracován. Protože jsou data v těchto databázích uložena jako relační, je poté nasazen mapper těchto dat na objektová data. Objektová data pak používá LINQ. Na tomto principu je výhodné především to, že s daty se pracuje jako s objekty.

Linq to SQL má následující výhody:

1. Odstínění od rozdílů implementací SQL jazyků pro databáze různých výrobců (Linq SQL dotazy sestaví sám).
2. Méně kódu – není nutné psát kód ADO.NET (otevřít spojení, uzavřít spojení, ...)
3. Deferred Execution – SQL dotazy prováděny dávkově, až je provedení nutné, série malých změn v datech nebo několik přidáných instancí do kolekce se promítne do databáze v podobě jednoho dotazu.

Nevýhodou Linq to SQL je pak skutečnost, že je do instance třídy možno vložit cokoli, co je správné z hlediska aplikace (typ dat, rozsah hodnoty), zároveň však vstup nemusí vyhovovat integritním omezením nastaveným v tabulce (dvě stejné hodnoty primárního klíče, rozsah hodnoty a další chyby). Tato chyba se projeví výjimkou, až se SQL dotaz založený na změnách v objektu začne provádět.

4 Systémy pro malé a střední weby

V této kapitole se budu zabývat především CMS systémy pro malé a střední projekty, či společnosti.

4.1 Požadavky na CMS systémy

Základní funkcionality pro tento druh webů rozeberu v následujících odstavcích.

Grafika

Při realizaci každého webu je grafika zásadní. Toto si myslí 46 % uživatelů Internetu [19]. Pokud bude váš web opatřen grafikou, která uživateli takzvaně "nesedne", existuje tedy velké riziko, že se rozhodne podívat po vaší konkurenci.

Grafika podle řady paradigmat ze světa byznysu určuje identitu projektu, či firmy. Ze sociologického hlediska má grafika na návštěvníky účinek v tom smyslu, že když váš web navštíví a zapamatují si co nejvíce věcí z této oblasti, tak pokud vás navštíví znovu, velice rychle si vzpomenou na návštěvu poslední.

Vyhledávání

Prostředí pro zadání vstupu vyhledávání může být realizováno například jednoduchým formulářem s jedním polem pro zadání hledaného textu, který je hledán ve všech příspěvcích. V rozsáhlejších systémech je ale implementován navíc rozšířený formulář, který umožňuje hledání různě specifikovat (jméno autora, časové období, typ obsahu, klíčová slova, jazyková verze a další). Samotné prohledávání dat pak probíhá obvykle dvěma způsoby.

Prvním je výběr z databáze s užitou funkcí LIKE `"*Text*"`. Tato cesta ale není moc efektivní, a to i navzdory případné indexaci prohledávaných atributů, protože indexace textu je sama o sobě náročná.

Druhou cestou je řešení jedním z kompromisů, jež jsou založeny jen na procházení klíčových slov.

Tato klíčová slova jsou buď zadána přímo uživatelem, nebo jsou výsledkem práce programu (nebo uložených procedur). Tato práce obvykle probíhá na pozadí, a v době, kdy je na serveru minimální provoz. Program (nebo uložená procedura) provede dotaz, který sestaví seznam slov, jež jsou obsažena v textech daného objektu.

Tento seznam slov je pak opatřen hodnotou primárního klíče daného objektu a identifikací jeho typu. Poté je celá kolekce dat vložena do databáze. Následné hledání pak probíhá nejprve v tomto vazebním slovníku, z něhož jsou získány identifikátory objektů, které jsou poté použity k získání vlastních objektů. Byť se může zdát, že hledání ve dvou tabulkách je neefektivní, přináší toto řešení velkou úsporu času při vykonávání vyhledávacích operací.

Správa kategorií a menu

Menu obsahuje odkazy na kategorie. Jedná se o základní prvek pro pohyb v datech v systému. Může být 1 úroňové, které je vhodné u jednodušších systémů a menšího

množství dat, jako například blogy. U rozsáhlejších a profesionálnějších systémů či projektů je nezbytné, aby menu bylo více úrovněvé. Vzhledem k tomu, že realizace víceúrovňového menu je jen mírně složitější než realizace toho jednoúrovňového, mělo by menu být implementováno menu více úrovněvé. Padne tak jedno ze zásadních omezení funkčnosti systému.

Editace a správa různého obsahu

Základem každého systému je zajistit kromě uživatelské strany systému i prostředí pro správu daného obsahu. Toto řešení by mělo ctít zásady uživatelsky komfortní práce s daty, kterou jsem již popsal v kapitole 2.3.1.

Vizuální editor pro úpravu textu

Pro pohodlné formátování textu objektů v systému je vhodné využít WYSIWYG editor pro editaci textu ve formátu HTML. Takovýto editor totiž oprošťuje uživatele od nutnosti znát HTML syntaxi, čímž se práce zjednoduší hlavně pro laiky.

Multijazykovost obsahu a lokalizace rozhraní

Multijazykovost obsahu a systémového prostředí je stále více používaná a žádaná funkcionality. V případě prezentací jednotlivců je tento požadavek méně častý, ale v případě prezentací firem je tato funkcionality nezbytností, protože potřebují při svém podnikání často komunikovat se zahraničím. Vícejazyčné systémové prostředí navíc pomáhá danému systému, aby se rozšířil i za hranice země svého původu.

Správa uživatelských rolí a práv

Řízení modulů administrace na základě systému rolí striktně vymezuje uživateli funkcionality, které smí a které nesmí využívat. Administrátor může řídit veškerá data systému. Naproti tomu řadový uživatel může například pouze přidávat příspěvky. Pro CMS systém jsou vhodné alespoň 3 role - administrátor, běžný uživatel, anonymní. Bez této funkcionality by si každý uživatel v systému mohl dělat cokoli, což by užitou hodnotu systému degradovalo.

Podpora pro RSS výstupy

RSS kanály (RSS feeds) zpřístupňují uživatelům obsah webu prostřednictvím funkce Informační kanály (Internet Explorer), podobné v jiných webových prohlížečích (Firefox, Chrome, Opera) nebo software označovaný jako čtečka informačních kanálů (například RSS Bandit, ...). Weby s RSS kanály pomáhají oslovit širší skupinu uživatelů v podstatně kratším čase než weby bez této funkcionality.

Například si můžeme vzít zkušenost ze života: uživatel objeví nový web, který ho zaujme svým obsahem. Autor svůj web přestane nějakou dobu obnovovat (je například na služební cestě), přičemž uživatelé z důvodu neaktuálnosti o jeho web ztratí zájem. Pokud by měl tento web RSS kanály, uživatelé si je budou moci nastavit pro odběr. Jakmile se web začne znovu aktualizovat, uživatelé velmi brzy získají informaci o aktualizaci.

Vazba mezi uživatelem a webem je takto zachována.

Kalendář akcí

Funkcionalita Kalendář umožňuje uživateli pohodlné procházení příspěvků přidaných v průběhu daného dne nebo časového období. Kalendář, například v podání Google, může také nahrazovat informační zprávy o plánovaných setkáních, čímž může být obsah webu zpřehledněn pro sdělení zásadnějších informací.

Diskuze

Diskusní fórum umožňuje uživatelům vyjádřit svůj názor na konkrétní příspěvek v systému. Častěji bývají příspěvky koncipovány ve formě stromu, někdy v chronologické jednoúrovňové podobě. Jednoúrovňová podoba diskuze nutí návštěvníky k uvádění jmen diskutujících, na které chtějí reagovat, podstatně častěji než stromová struktura.

Kontaktní formulář

pro hlášení chyb a nedostatků v souvislosti s webem působí z hlediska uživatele lépe než odkaz na emailovou adresu. Je třeba mít na paměti, že ne každý počítač musí mít nainstalovaného poštovního klienta nebo nemusí mít povoleno komunikovat po síti SMTP protokolem. Navíc kontaktní formulář předepisuje strukturu sdělovaných informací podstatně lépe než text emailu a umožňuje tak automatizovanou a tedy snazší evidenci tohoto typu dat.

Pokud se bude jednat o web menší společnosti, bude vhodná i implementace dalších funkcionalit, jako je například:

Administrativní zóna

Administrativní zóna uživateli poskytuje přístup k firemnímu emailu, datovému úložišti, intranetu nebo administrativnímu rozhraní podnikového informačního systému. Přístup je poskytován pouze proti jménu a heslu, které projde kontrolou autentifikačního mechanismu.

Elektronický obchod a bezhotovostní platby byl již rozebrán v kapitole 2.5 v oddílu **Elektronický obchod**.

Řízení reklamních ploch

Umísťování reklamy je nejjednodušší způsob, jak na Internetu vydělávat peníze. Další způsoby, jako například komerční články, či sdělení již tak efektivní nejsou.

Proto je vhodné, aby publikační systémy měly alespoň systém pro řízení reklamních ploch. Autoři systémů, kteří to s budoucností svého systému míní opravdu vážně by mohli zvážit vývoj vlastního reklamního systému.

Systém pro řízení reklamních ploch se stará o řízení míst na portálu, kam je možno reklamy umísťovat. Tento systém pak bývá propojen s reklamním systémem, buď to vlastním

nebo třetí strany, jako je například Sklik, Google AdSense a jiné.

Reklamní systém

Cílem reklamního systému je využívat alokované reklamní plochy a v nich zobrazovat sjednané reklamy od inzerentů.

Reklamní systém by měl mít především tyto funkcionality:

1. Evidence zákazníků
2. Evidence vypsaných kontraktů a zajištění jejich plnění (vypisování reklamních bannerů do stránky)
3. Napojení na elektronický platební systém (například: PayPal)
4. Funkce pro plánování reklamních kampaní
5. Funkce pro podporu cílení reklamy (možnost inzerovat jen v regionu namísto celoplošné reklamy, ...)
6. Evidence počtu zobrazení, prokliků a dalších indikátorů úspěšnosti reklam

Podpora SEO optimalizace

SEO optimalizace je optimalizací stránek pro vyhledávače. Jedná se o soubor konceptů, které navrhuji, jak koncipovat své stránky z pohledu zdrojového kódu i informačního obsahu, aby tento obsah byl pro vyhledávač co nejsnáze automatizovaně zpracovatelný a katalogizovatelný. Pozitivním výsledkem SEO je zisk přední pozice ve výsledku vyhledávání, pokud jsou hledána klíčová slova na firemní stránce.

Anketní systém

Často může být výhodné získat od uživatele zpětnou vazbu, jak vnímá obsah portálu nebo sdělované informace. Nejjednodušším nástrojem, jak tuto zpětnou vazbu získat, je anketa.

Nejčastější realizace anket je prováděna tak, že je zadána otázka, na kterou je možné zvolit pouze jednu odpověď z předdefinovaného souboru přípustných odpovědí. Poté, co uživatel zvolí odpověď a klikne na tlačítko Odeslat, je odpověď i s IP adresou počítače odeslána do databáze. Tím je zaručeno, že z každého počítače lze hlasovat jen jedenkrát.

Jak jsem nyní popsal, proces získání dat od návštěvníka je velice jednoduchý především pro návštěvníka, protože nemusí na otázku odpovídat ručně, což šetří jeho čas, a je tak velice vysoká šance, že na anketu odpoví.

4.2 Alternativy pro chybějící funkcionality

V této podkapitole uvedu několik náhradních produktů třetích stran pro některé žádané funkcionality CMS systémů. Tato kapitola bude pojata spíše z technického než z uživatelského pohledu.

Vyhledávání

je možno řešit zjednodušeným formulářem, který bude data odesílat v podobě, jež je přijatelná vyhledávacím strojem.

Diskuze

Absenci diskusního fóra lze nahradit následujícími produkty:

1. plugin pro sociální síť (Facebook, Twitter, Google Plus a další)
2. jiný CMS systém zaměřený na diskusní fóra, například PHP BB. Napojení je často realizováno v podobě modulu hlavního CMS systému, který lze volitelně vypnout, či odinstalovat. Hlavní CMS systém pak posílá příkazy systému diskusního fóra
3. jednodušší funkcionality třetí strany, například: BlueBoard)

Anketní systém

Při absenci anketního systému je možné použít externí službu, jako například: BlueBoard.

5 Realizace systému SimplePublishCMS

5.1 Vize realizace

Hlavním cílem je vyvinout publikační systém především pro malé a střední projekty v prostředí technologie ASP.NET MVC3. S ohledem na tento požadavek bude realizace zaměřena na to, aby poskytla těmto projektům zázemí pro publikaci obsahu v podobě základních nástrojů pro práci s tímto obsahem.

Malé a střední weby považují za klíčové především tyto funkcionality:

1. administrace uživatelů a rolí
2. tvorba, administrace a publikace obsahu (nejčastěji články, odkazy, skupiny, RSS kanál, soubor ke stažení)
3. uživatelsky intuitivní procházení obsahu (například vyhledávání podle různých požadavků)

Protože jedním ze základních cílů publikačních systémů je zajistit oddělení obsahu od způsobu a formátu jeho zobrazení, bude za tímto účelem zvolena pro realizaci technologie šablon. Konkrétně se jedná o systém Razor, který s ASP.NET MVC3 výborně spolupracuje.

5.1.1 Vstupy

Systém si bude uchovávat data vložená uživateli, mezi která patří:

1. informace o obsahu (články, odkazy, skupiny, RSS kanály, soubory ke stažení), jako jsou datum vytvoření a zveřejnění, příznak publikace do portálu a RSS, autorství a další
2. Seznam uživatelů
3. Konfigurační volby systému a jednotlivých uživatelů
4. Seznam dostupných jazyků
5. Slovník klíčových slov
6. Menu a jejich položky
7. Číselníky, položky číselníků
8. Seznam IP adres
9. Názory a doporučení návštěvníků

Mezi vstupy patří kromě dat také:

1. šablony a kaskádové styly definující formu výstupu dat pro uživatele
2. obrázky, videosoubory, audiosoubory a další obsah

5.1.2 Výstupy

Výstupem systému budou šablony, které budou doplněny o data od uživatelů, kaskádové styly a grafiku. Výstup CMS systému je možno rozdělit do dvou základních obecných typů a to (X)HTML stránky a RSS výstupy. Stejně tomu bude i v případě SimplePublishCMS.

5.1.3 Základní organizace CMS systému

Netriviální CMS systémy bývají téměř bez výjimky koncipovány v následujících dvou hlavních blocích:

1. Uživatelská část (Front End) – tato část je volně přístupná návštěvníkům bez přihlášení
2. Administrativní část (Back End) - tato část je přístupná uživatelům pouze po přihlášení jejich uživatelským jménem a heslem. Umožňuje uživatelům se do určité míry (omezené přidělenými právy) podílet na tvorbě obsahu webu a zpřívětivnit si daný web volbami osobního nastavení, které jim povolí administrátoři (jiný CSS styl, ...).

Pro uživatele, kteří chtějí získat účet do CMS systému, je k dispozici registrační formulář. Po jeho vyplnění dostane uživatel emailovou zprávu s odkazem pro konečné potvrzení uživatelských informací. Často je nastavena maximální doba od odeslání, do které lze účet aktivovat. Pokud uživatel účet neaktivuje, účet propadne.

Takto bude organizován i SimplePublishCMS. Možnost registrace nových uživatelů bude v systému řízena konfigurační volbou. Systém bude takto nastaven, aby vyhověl požadavkům firem i menších webů.

5.2 Use-Case model

5.2.1 Use-Case diagram

V obrázku 8 jsou uvedeny případy užití pro role SuperUser a Návštěvník. V obrázku 9 je pak uveden seznam případů užití pro roli ContentContributor. Use-case model, který je zde uveden, popisuje jen nejzákladnější případy užití v systému.

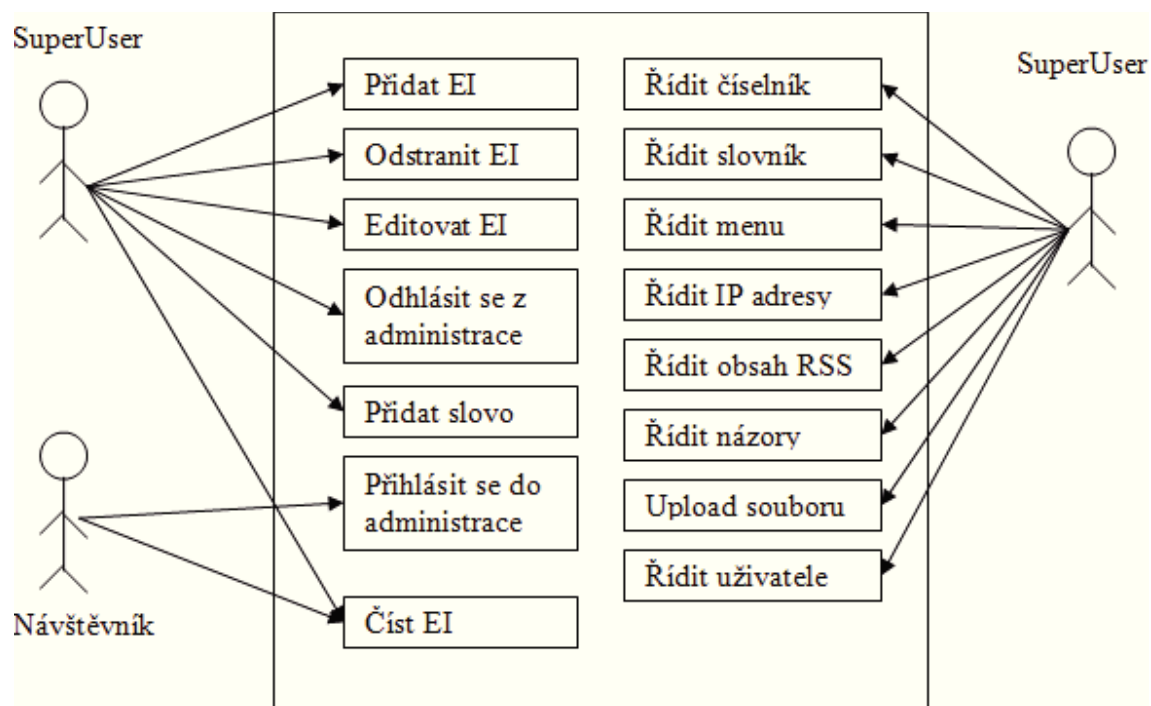
EI = Executing Item, spouštěcí položka

5.2.2 Seznam událostí a reakcí

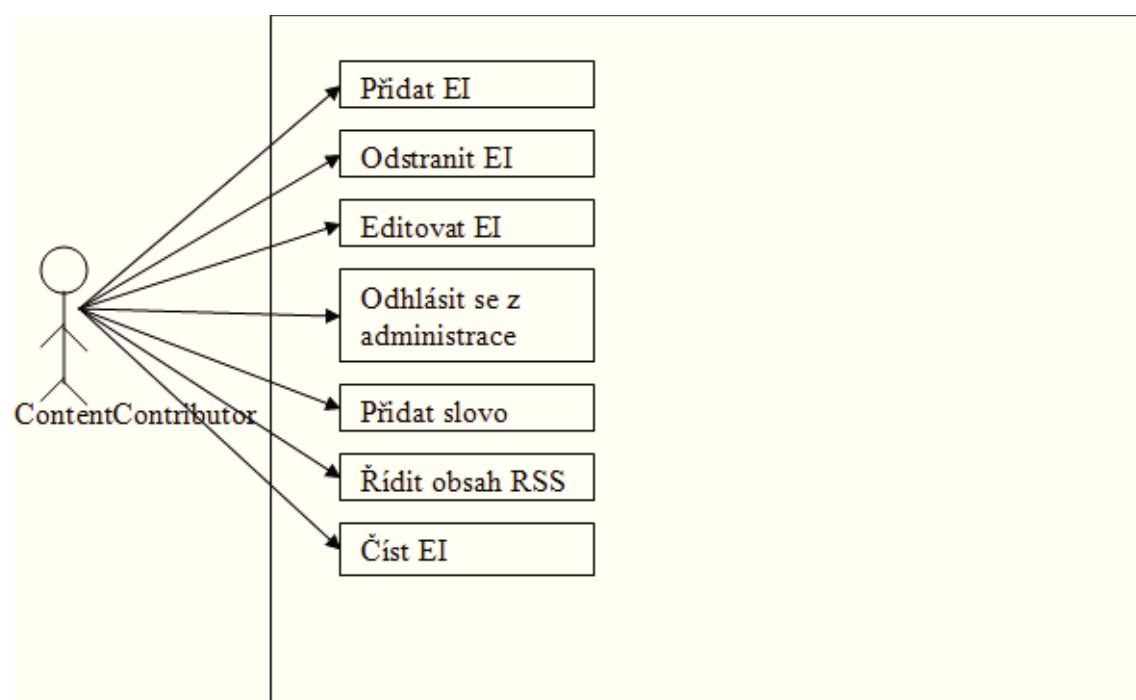
Seznam událostí a reakcí je uveden v tabulce 1.

5.2.3 Seznam aktérů

Na obrázku 10 je uveden seznam aktérů a dědičnost mezi rolemi.



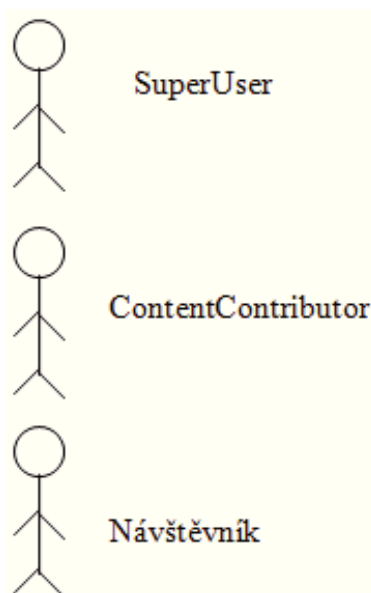
Obrázek 8: Use-case diagram: aktivita: SuperUser a Návštěvník



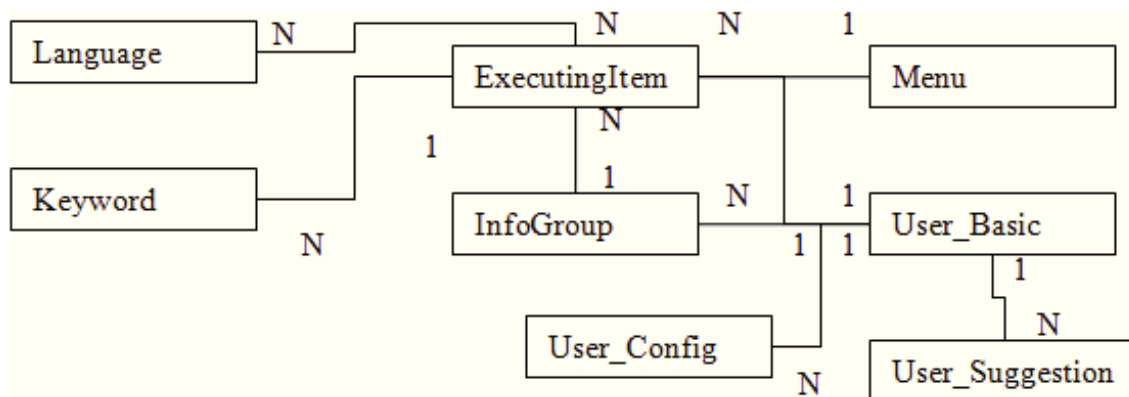
Obrázek 9: Use-case diagram: aktivita: ContentContributor

#	Událost	Reakce	Aktér
1	Přidat EI	Vloží EI s požadovanými informacemi	SU,CC
2	Odstranit EI	Dotáže se a při souhlasu zruší EI	SU,CC
3	Editovat EI	Vloží EI se změněnými informacemi	SU,CC
4	Odhlásit se z administrace	Odhlásí uživatele z administrace	SU,CC
5	Přidat slovo	Při přidávání EI vloží klíčová slova do slovníku	SU,CC
6	Přihlásit se do administrace	Po zadání správných údajů je uživatel přihlášen	N
7	Číst EI	Zobrazí uživateli celou EI	SU,CC, N
8	Řídit číselník	Přesune uživatele na panel řízení číselníků	SU
9	Řídit slovník	Přesune uživatele na panel řízení slovníků	SU
10	Řídit menu	Přesune uživatele na panel řízení menu	SU
11	Řídit IP adresy	Přesune uživatele na panel řízení IP adres	SU
12	Řídit obsah RSS	Umožní uživateli řídit obsah RSS kanálu	SU, CC
13	Řídit názory	Přesune uživatele k řízení uživatelských názorů	SU
14	Upload souboru	Umožní uživateli uploadovat soubor na server	SU
15	Řídit uživatele	Přesune uživatele na panel řízení uživatelů	SU

Tabulka 1: Seznam událostí a reakcí



Obrázek 10: Aktéři a dědičnost



Obrázek 11: Class diagram (zjednodušený)

5.2.4 Popis událostí a reakcí

Popis událostí a reakcí je uveden v příloze **Rozbor případů užití Use-Case modelu**, která je umístěna na příloženém CD. Tato příloha se stejně jako Use-Case diagramy zaměřuje především na základní reakce a funkcionality v systému. V této kapitole budou zmíněny jen vstupní kontroly systému.

Vstupní kontroly

Tyto kontroly probíhají před provedením každé následující akce. Jejich cílem je prozkoumat oprávněnost daného uživatele na požadovanou akci. Při této realizaci byla využita technika CAS, která omezuje přístup do kódu programu nepovolaným. Další testy se týkají kontroly IP adresy, jestli není na seznamu zamítnutých, ohledně přihlášených uživatelů se provede podobná kontrola.

5.2.5 Třídní diagram

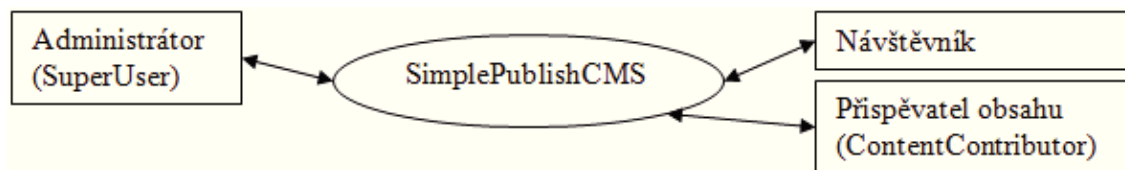
Na obrázku 11 je uveden diagram tříd. Tento diagram je zjednodušený a zachycuje jen nejdůležitější datové objekty systému. Podrobnější popis datové vrstvy naleznete v příloze **Rozbor datové vrstvy**, která je uložena na příloženém CD.

5.3 Technická specifikace

5.3.1 Uživatelské role v systému

Administrátor (SuperUser)

Administrátorská role dovoluje svému držiteli provádět veškeré operace se systémem. Může spravovat uživatele (včetně blokování), slovník klíčových slov, číselníky, blokovat IP adresy. Dále, pokud má přidělenou i roli ContentContributor, může také pracovat s veškerými obsahovými daty v systému.



Obrázek 12: Kontextový diagram

Oblast	Technologie
Databázový stroj:	Microsoft SQL Server 2008
Technologie přístupu k datům:	ORM - Entity Framework; Linq
Operační systém:	Microsoft Windows Server 2008 R2
Technologie IS:	Microsoft ASP.NET MVC 3 (C#)
Technologie šablon:	Razor View Engine
Technologie na straně klienta:	JavaScript, jQuery, TinyMCE
Ostatní technologie:	.NET Framework, HTML, CSS

Tabulka 2: Výčet technologií SimplePublishCMS

Příspěvatel obsahu (ContentContributor)

Tato role umožňuje uživateli pracovat s veškerými obsahovými daty v systému jichž je autorem a oproti administrátorovi nemůže řídit systémové číselníky. Do slovníku klíčových slov může uživatel slova pouze přidávat.

Návštěvník

Tato „role“ umožňuje uživateli pouze číst zveřejněný obsah z uživatelské strany portálu. Uživatel může pouze posílat nápady nebo výtky k činnosti portálu prostřednictvím kontaktního formuláře. Uživatel není veden v seznamu uživatelů systému.

5.3.2 Kontextový diagram

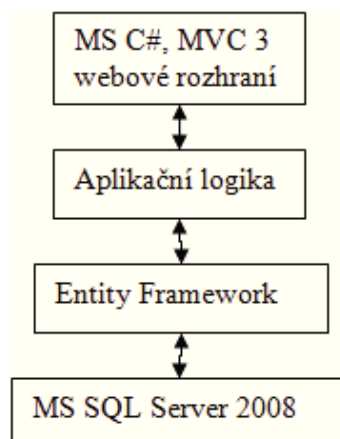
Na obrázku 11 je uveden kontextový diagram znázorňující okolí systému.

5.3.3 Výčet technologií

Seznam technologií je uveden v tabulce 2.

5.3.4 Použitelnost systému

- Systém má jednoduchý HTML výstup, čitelný v nejzastoupenějších webových prohlížečích (Internet Explorer, Mozilla Firefox, Chrome, Opera).
- Systém čerpá při návrhu uživatelského rozhraní z obvyklých konvencí, takže bude pro uživatele snadný k pochopení.



Obrázek 13: Schéma architektury

5.4 Bezpečnost systému

Bezpečnost systému má následující rysy

- Data jsou uchovávána v databázi s přístupem přes administrativní účet chráněný heslem. Slabinu může představovat prolomení hesla FTP účtu a přečtení souboru **web.config**, ve kterém je heslo a jméno pro přístup do DB uloženo v nechráněné podobě.
- Hesla uživatelů jsou kódována a uložena v samostatné MS SQL databázi (MS SQL Sever Compact) na souborovém úložišti.
- Klientská strana systému bude přístupná návštěvníkům volně, tedy bez hesla.

5.5 Schéma architektury

Na obrázku 13 je uvedeno blokové schéma architektury.

5.6 Další technologie použité v realizaci

5.6.1 JavaScript

Java Script byl poprvé uveden v roce 1995 v Netscape Navigatoru od Netscape Communications, v roce 1996 představil Microsoft ve svém prohlížeči Internet Explorer 3 podobnou implementaci nazvanou JScript. [7]

Java Script je standardizovaný multiplatformní programovací jazyk interpretovaný na straně klienta, byl vyvinut s cílem řešit dynamiku HTML stránek. Java Script je syntakticky blízký Javě, C nebo C++. Dalšími rysy Java Scriptu jsou case-sensitivita, je objektově orientovaný, beztřídní, prototypový, dynamicky typovaný. Hlavním rysem je

skutečnost, že je omezen pouze na rozsah prostředí prohlížeče (například neumí pracovat se soubory na pevném disku) a je závislý na rozdílnostech implementací jednotlivých prohlížečů, což při vývoji přináší často problémy.

Java Script je schopen: [22] [7]

- pracovat a měnit DOM model HTML dokumentu,
- pracovat s prohlížečem (průchod a čtení historie, příkaz tisk, nastavit jako domovskou stránku, ...)
- počítat jednodušší matematické úlohy,
- zpracovat události plynoucí z činnosti myši a klávesnice,
- Spolupracovat s Java Applty a COM objekty,
- pracovat s Cookies daty,
- zpracovávat data formuláře a pracovat s formulářem (kontroly formátu dat v polích)

Java Script je možné umístit do HTML stránky třemi způsoby:

1. Do hlavičky HTML - odkaz na externí

```
<script type="text/ javascript " src="externi . js"></script>
```

Výpis 3: Tag pro import JavaScript souboru do hlavičky HTML

2. V hlavičce HTML

```
<script type="text/ javascript ">
  <!-- kompatibilita pro staré prohlížeče
    alert (" Vítejte ");
    // Konec kompatibility -->
</script>
<noscript>
  Starý prohlížeč nepodporuje JavaScript, pořídte si novější.
</noscript>
```

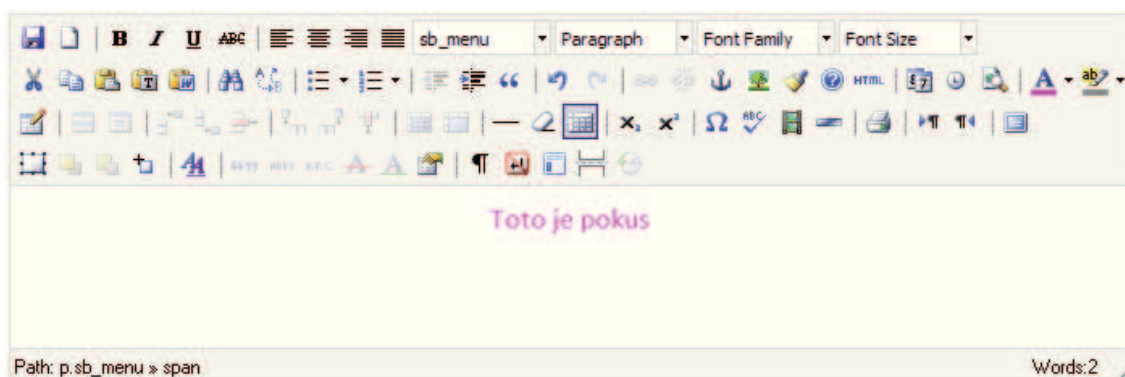
Výpis 4: Definice JavaScriptu v hlavičce HTML

3. Součástí tagu

```
<a href="javascript :Zprava();">Zpráva</a>
```

Výpis 5: JavaScript jako součást tagu

Stále častěji bývají v různých webových stránkách s oblibou nasazovány Java Scriptové frameworky (knihovny funkcí), například jQuery, Prototype, GWT, Dojo, Moo Tools a další. Právě o prvním jmenovaném se v následující kapitole zmíním blíže.



Obrázek 14: TinyMCE WYSIWYG HTML editor

5.6.2 jQuery

Jak jsem již zmínil, jQuery je JavaScriptový framework. Jedná se o nejčastěji používaný framework. jQuery je zaměřený především na zjednodušení práce s událostním modelem JavaScriptu a objekty v HTML dokumentu, respektive jejich vlastnostmi. Obsahuje také několik animací, které je možno uplatnit na prvky v HTML dokumentu.

Ukázka kódu jQuery je naznačena ve výpisu 6.

```
<script type="text/javascript">
  $(function () {
    $("#PublishDate").datetimepicker({
      showSecond: true,
      timeFormat: 'hh:mm:ss',
      dateFormat: "d.m.yy",
      numberOfMonths: 2
    });
  });
  $(document).ready(function () {
    ActivateTinyMCE("BriefDescription", "simple");
    OnOffSekce(["show1"]);
  });
</script>
```

Výpis 6: Ukázka kódu jQuery

5.6.3 TinyMCE

Autor: MoxieCode, <http://www.moxiecode.com>

Licence: GNU GPL

TinyMCE usnadňuje psaní textu a oproštuje uživatele od nutnosti znát HTML syntaxi, jedná se totiž o JavaScript knihovnu s WYSIWYG editorem HTML kódu.

5.6.4 Další technologie a produkty třetích stran

TimePicker

Autor: Trent Richardson, <http://trentrichardson.com/examples/timepicker/>

Licence: GNU GPL, MIT

Nadstavbový prvek pro jQuery. Jedná se výběrový prvek pro pole typu datum a čas (DateTime) . Jedná se Java Script knihovnu.

HtmlRemoval class

Autor: <http://www.dotnetperls.com> [16]

Licence: nelicencováno

Tato C# třída zajistí vyčištění textu od HTML tagů.

5.7 Přínos technologií

Nyní blíže rozeberu význam použitých technologií pro provedenou realizaci.

Microsoft SQL Server 2008 dodal realizaci výkonnou a stabilní platformu pro ukládání dat.

Entity Framework vývojáře oprostí od nutnosti implementovat objektově relační mapování. Na druhou stranu přinese určité zhoršení propustnosti datové vrstvy. Velkým pozitivem EF je také skutečnost, že dědičnosti rozkládá na tabulku zastupující rodičovskou třídu a tabulku zastupující následníka.

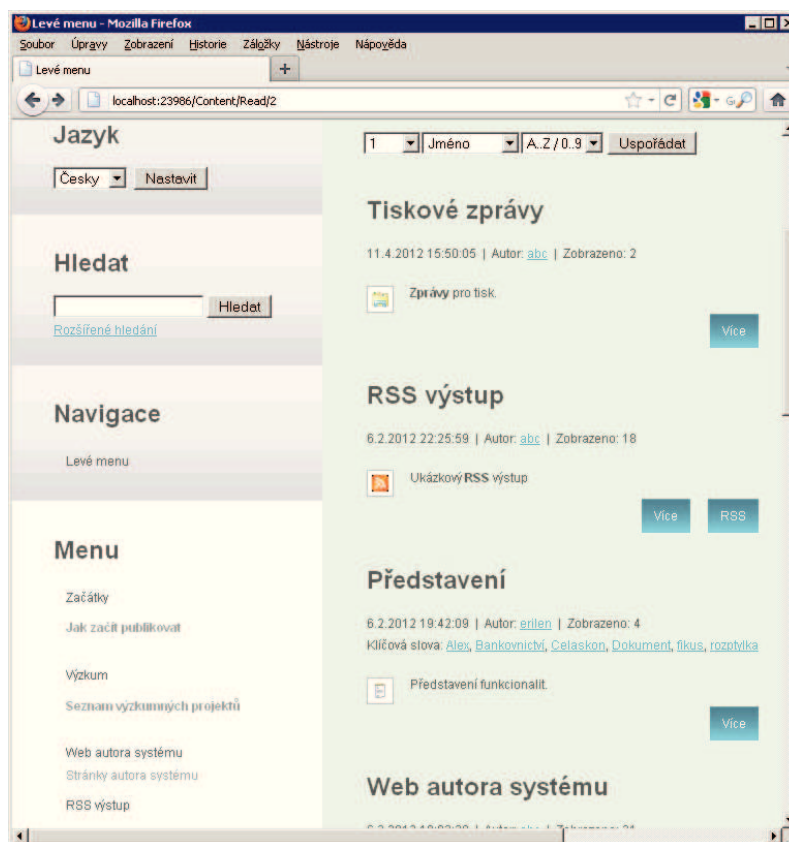
Při změnách ve struktuře, až budou lépe ve Visual Studiu implementovány podpůrné funkce, toto uspořádání omezuje změnu struktury jen na jednu tabulku. Změny struktury dat jsou v současnosti ve Visual Studio řešeny způsobem vše zrušit a znovu vytvořit, a to včetně dat.

Linq vývojáře oprostí od nutnosti připravit dotazy pro práci s databází, na druhou stranu přinesí vzhledem ke svému způsobu vnitřní realizace zhoršení propustnosti datové vrstvy.

Microsoft ASP.NET MVC 3 přineslo zásadní zpřehlednění struktury zdrojového kódu a pomocí rozsáhlé knihovny tříd přineslo také velké zjednodušení vývoje. Velmi pozitivně lze na MVC3 vnímat schopnost řešit celou řadu běžných chyb a nepříjemných drobností, které se musejí v některých jiných technologiích při vývoji řešit, například přijetí HTML kódu formulářem, kde by byl z důvodu bezpečnosti vhodnější prostý text.

Razor View Engine, coby šablonovací systém, vnesl do realizace webových aplikací požadovanou pružnost prezentační vrstvy. Usnadnil tak oblasti realizace související s touto vrstvou - zejména v oblasti implementace více kaskádových stylů v systému. Další zjednodušení vývoje přinesly vestavěné funkce pro generování HTML kódu (helpery), které částečně, zejména v jednodušších konstrukcích HTML kódu, oprostují programátora od nutnosti HTML kód psát celý ručně. Velkým pozitivem je také možnost definovat si podobné helpery i uživatelsky.

Technologie jQuery a JavaScript vylepšily flexibilitu realizace v mnoha oblastech, například v oblasti kontroly formulářů. Dále byla posílena flexibilita této realizace. TinyMCE ušetřil náročnou implementaci WYSIWYG editoru.



Obrázek 15: Prostředí systému: Procházení obsahu

5.8 Ukázky uživatelského prostředí systému

V této kapitole pro názornost uvedu jen několik screenshotů SimplePublishCMS, aby z nich bylo lépe patrné, čeho bylo realizací systému dosaženo.

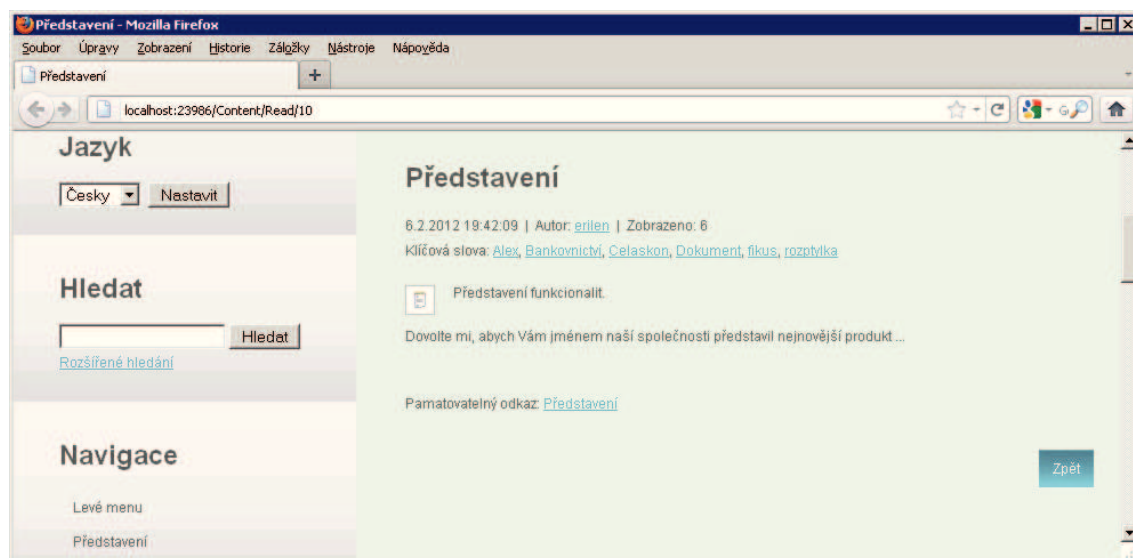
5.9 Administrace obsahu (strana přihlášeného uživatele)

V následujících bodech je uveden základní výčet rysů administrativního rozhraní systému:

- Pokud si uživatel přeje vybrat jiný jazyk dat, může tak učinit na administrativním panelu (stejným způsobem, jako v uživatelské části systému)
- Veškeré moduly jsou dostupné z administrativního menu
- Pokud uživatel potřebuje přistupit k podřízeným položkám, může tak učinit příkazem **Manage** (obrázek tabulky)
- K přidání položky slouží akce **Create** a **Create New**. Uživateli je vždy zobrazen formulář se všemi nastavitelnými daty.



Obrázek 16: Prostředí systému: Pohled na EI (Článek)

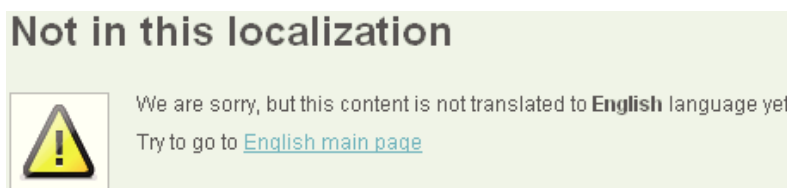


Obrázek 17: Prostředí systému: Pohled na úplný výpis EI (Článek)



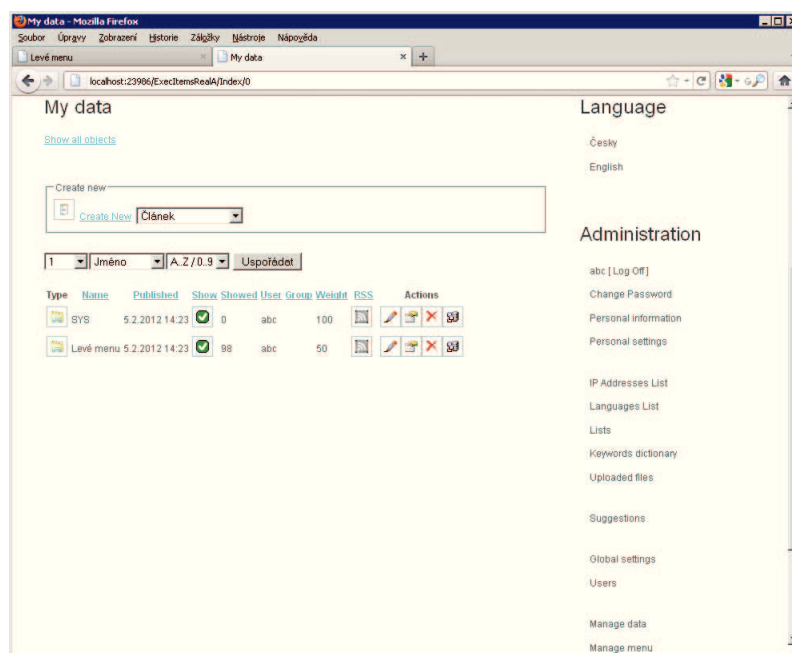
Obrázek 18: Prostředí systému: Pohled na EI (RSS výstup)

Obrázek 19: Prostředí systému: Výběrové pole pro datum a čas

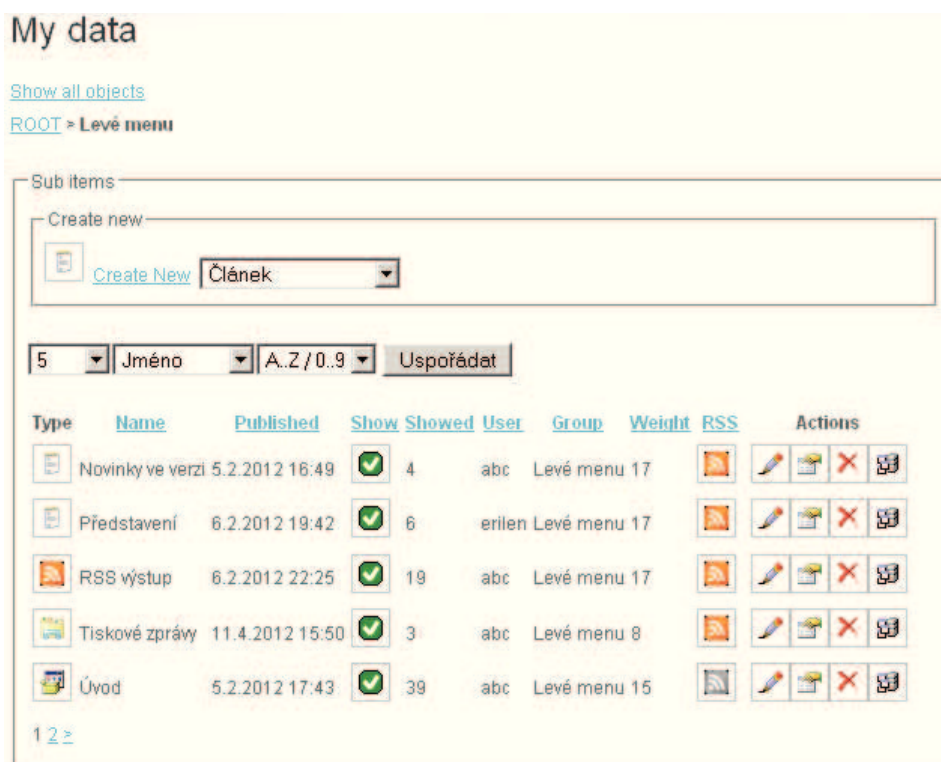


Obrázek 20: Prostředí systému: Informační zpráva systému (položka bez lokalizace)

- Pro úpravy položek slouží akce Edit (obrázek tužky). Uživateli je vždy zobrazen formulář s dosavadními hodnotami.
- Po potvrzení změn ve formulářích Create a Edit je provedena kontrola typů i integrity dat a když proběhne úspěšně, jsou data uložena do databáze. Pokud úspěšně neproběhne, je uživatel vyzván k nápravě chyb a postup se opakuje.
- K odstranění položky je určena akce Delete (obrázek červeného křížku). Po kliknutí na tuto akci je vždy uživatel požádán o potvrzení této akce.
- Ostatní akce jsou volitelné podle modulu. Jejich úkolem je většinou strategickou proměnnou nastavit do opačného stavu, například (Od)Blokovat, (Ne)Viditelnost nebo provést jednorázovou operaci, jako nastavit na výchozí a další.



Obrázek 21: Prostředí systému: Administrace



Obrázek 22: Prostředí systému: Administrace spustitelných položek (EI)


My data

[ROOT](#) > [Levé menu](#) > **Představení**

Actions



Keywords

Alex, Bankovníctví, Celaskon, Dokument, fikus, rozptylka, 

Localizations

English

Menu items

[Create New](#)

Obrázek 23: Prostředí systému: Administrace dalších možností EI (článek)

Menu

Začátky

Jak začít publikovat

Výzkum

Seznam výzkumných projektů

Web autora systému

Stránky autora systému

RSS výstup

Ukázkový RSS výstup

Obrázek 24: Prostředí systému: Pohled na jeden z modulů (Menu)

Edit

Menu item

Menu

Executing item

Visibility

Show type

Weight

☒ Spread object

[Back](#)

Obrázek 25: Prostředí systému: Administrace : Akce Edit (položka menu)

Delete

Are you sure you want to delete this?

List

Name
UserNumList
Description
UserNumList
Group
SYS
System
No

| [Back to List](#)

Obrázek 26: Prostředí systému: Administrace : Akce Delete (číselník)

6 Zhodnocení systému SimplePublishCMS

V této kapitole zhodnotím realizaci systému SimplePublishCMS a blíže nastíním cesty, kam by mohl vývoj tohoto systému směřovat.

6.1 Problémy při realizaci systému

Při vývoji systému se objevilo několik náročnějších úloh, které bylo nutné vyřešit.

Nejnáročnější úlohou bylo propojení šablony a editoru TinyMCE. Zde jsem si nechal poradit od blogu jednoho z vývojářů, protože se objevila slabší podpora integrace knihovny TinyMCE.

Další náročnější částí realizace byla integrace kontroly IP adresy. V této části se jednalo o nalezení vhodného místa pro spuštění této kontroly. Jednalo se o událost **OnActionExecuted**.

Ostatní zajímavé části realizace jsou dále rozebrány ve stejnojmenné kapitole, která není součástí vytištěných příloh a je uložena na přiloženém CD.

6.2 Možnosti dalšího rozvoje systému

Jak zmiňuji v kapitole A.1 o technologii HTML, další vývoj systému bude mimo jiné těžit z novinek, které přinese **HTML 5**. Vývoj by tedy měl primárně směřovat k implementaci rysů tohoto budoucího standardu. Při dalším vývoji systému by bylo vhodné systém přizpůsobovat i pro mobilní zařízení. Dále by bylo také vhodné využít technologie AJAX.

Druhou oblastí, kam by vývoj mohl směřovat, je dokončení jazykového překladu celého prostředí systému do více jazyků, vytvoření prostředí pro podporu externích modulů a následná implementace dalších modulů. Mohlo by se jednat o ankety, diskusní fórum, podporu akcí a fotogalerii.

Následně by bylo vhodné se také orientovat na vybudování rozhraní pro spolupráci se sociálními sítěmi a rozhraní pro vzájemné propojení různých instancí systému napříč Internetem.

Čtvrtou oblastí, kam by mohl vývoj systému směřovat, je podpora organizace seznamu produktů včetně realizace Nákupní košík a implementace vlastního subsystému pro řízení reklamního prostoru. V souvislosti s tímto cílem bude třeba také zajistit rozhraní pro komunikaci s bankovními servery, což si vyžádá aplikaci nejpřísnějších postupů a nástrojů zabezpečení na všech úrovních takto provozovaného systému, od hardware po software.

7 Závěr

Cílem této bakalářské práce bylo, kromě popisu technologií a analýzy potřeb a požadavků na současné CMS systémy, provést také vývojový proces ukázkového CMS systému. Při realizaci bylo použito technologií, které jsou v současné době řadou profesionálů doporučovány jako jedny z nejperspektivnějších.

V úvodu byly zmíněny obecné vlastnosti CMS systémů, funkcionality, které jsou po nich často žádány a nejpoužívanější technologie pro jejich realizaci. Implementace systému se inspirovala vlastnostmi nejpoužívanějších open source CMS systémů, které jsou uvedeny v příloze B.

V souladu se základním požadavkem - vytvořit publikační systém pro malé a střední projekty, vznikl procesem implementace CMS systém SimplePublishCMS, který je snadno pochopitelný, má intuitivní uživatelské prostředí a poskytuje plnohodnotné základní prostředky pro běžně zpracovávaná data. Systém SimplePublishCMS je způsobilý k nasazení v reálných aplikacích. Zejména v aplikacích menších webů.

Vzhledem k jednoduchosti vývoje v technologii ASP.NET MVC3 v porovnání s ASP, ASP.NET WebForms nebo ostatními uváděnými technologiemi a s ohledem na skutečnost, že MVC3 staví na moderních, mladých a dynamicky se rozvíjejících technologiích, o čemž svědčí i rychle rostoucí kvalitní blogy vývojářské komunity, lze říci, že technologie ASP.NET MVC3 má před sebou velkou budoucnost.

8 Reference

- [1] FREEMAN, Adam a SANDERSON. *Pro ASP.NET MVC 3 framework. 3rd. ed.* New York: Apress, c2011, 824 s. The expert's Voice in.NET. ISBN 978-1-4302-3404-3.
- [2] FOWLER, Martin. *Patterns of enterprise application architecture.* Boston: Addison-Wesley, c2003, 533 s. ISBN 03-211-2742-0.
- [3] Microsoft Developer Network. *Microsoft Developer Network* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.msdn.com>
- [4] MICROSOFT CORPORATION. Microsoft ASP.NET. *Microsoft ASP.NET* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.asp.net>
- [5] MICROSOFT CORPORATION. *Tutorials: Official Microsoft Site* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.asp.net/mvc/tutorials>
- [6] PUŠ, Petr. *Poznáváme C# a Microsoft .NET.* In: *Poznáváme C# a Microsoft .NET* [online]. 2006 [cit. 2012-04-25]. Dostupné z: <http://www.zive.cz/autori/sc-44/default.aspx?author=405>
- [7] JANOVSKEÝ, Dušan. *Jak psát web.* *Jak psát web* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.jakpsatweb.cz>
- [8] RADECKÝ, Michal. *Redakční a informační systém pro WWW portály.* VŠB-TU Ostrava, 2002. Diplomová práce. VŠB-TU Ostrava. Vedoucí práce SZTURC, Roman.
- [9] SANS INSTITUTE [online]. 2012 [cit. 2012-04-25]. Dostupné z: http://www.sans.org/reading_room/whitepapers/windows/microsoft-dotnet-overview_292
- [10] GALLOWAY, Jon. *Jon Galloway Blog* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://weblogs.asp.net/jgalloway/>
- [11] WILSON, Brad. *Brad Wilson* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://bradwilson.typepad.com/blog/>
- [12] HANSELMAN, Scott. *Scott Hanselman* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.hanselman.com/blog/>
- [13] DATABASE RESEARCH GROUP, VŠB-TU Ostrava. *Database Education* [online]. VŠB-TU Ostrava, 2012 [cit. 2012-04-25]. Dostupné z: <http://dbedu.cs.vsb.cz>
- [14] BONSANGUE, Marcello. Educational web. In: *Marcello M. Bonsangue* [online]. 2005 [cit. 2012-04-25]. Dostupné z: http://www.liacs.nl/marcello/CBSE/cbse_10a.ppt
- [15] *The CMS Matrix - cmsmatrix.org - The Content Management Comparison Tool* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.cmsmatrix.org/>

- [16] *Dot Net Perls* [online]. 2012 [cit. 2012-04-25]. Dostupné z: <http://www.dotnetperls.com/>
- [17] LACKO, Ľuboslav. *PHP a MySQL: hotová řešení. 1. vyd.* Brno: CP Books, 2005, 299 s. ISBN 80-251-0397-8.
- [18] TIOBE SOFTWARE. *TIOBE Software: TIOBE Index* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [19] NETDIRECT S.R.O. MediaCentrik [online]. 2012 [cit. 2012-04-28]. Dostupné z: <http://www.mediacentrik.cz/>
- [20] ALTOS SOFTWARE S. R. O. ALTOS [online]. 2012 [cit. 2012-04-28]. Dostupné z: <http://www.altos.cz/>
- [21] Publikáční systém WebGet [online]. 2012 [cit. 2012-04-28]. Dostupné z: <http://www.webget.cz/>
- [22] World Wide Web Consortium (W3C) [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.w3.org/>
- [23] ORACLE. *Oracle Technology Network for Java Developers* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.oracle.com/technetwork/java/index.html>
- [24] *The Perl Programming Language* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.perl.org/>
- [25] *The Comprehensive Perl Archive Network* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.cpan.org/>
- [26] *Ruby Programming Language* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.ruby-lang.org/en/>
- [27] *Ruby on Rails* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://rubyonrails.org/>
- [28] *PHP: Hypertext Preprocessor* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.php.net/>
- [29] VRÁNA, Jakub. *PHP triky - Weblog o elegantním programování v PHP pro mírně pokročilé* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://php.vrana.cz/>
- [30] *Python Programming Language - Official Website* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.python.org/>
- [31] *Django Project* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <https://www.djangoproject.com/>
- [32] *Standard ECMA-334* [online]. 2012 [cit. 2012-04-26]. Dostupné z: <http://www.ecma-international.org/publications/standards/Ecma-334.htm>

A Další technologie použité v realizaci

V této příloze rozebírám podrobněji další technologie využití ve své realizaci, které jsem neuvedl v kapitole 3.9.

A.1 HTML

HyperText Markup Language je značkovací jazyk pro hypertextové dokumenty v prostředí WWW, který vychází z univerzálního značkovacího jazyka SGML a byl uveden v neformální specifikaci poprvé v roce 1990 Timem Bernesem Lee (fyzik v ústavu jaderné a částicové fyziky CERN).

Tento jazyk určuje rozvržení dat v dokumentech a jejich vzájemné propojení. Jeho syntaxe se skládá z párových (``/``) a nepárových (`
` / `</>`) značek.

Mezi párovými značkami se obvykle nachází další text, například `Centrum`. Téměř všechny značky mají také atributy (například: `title`, `alt`, `src`, `style`, ...). Jejich hodnoty mají být podle specifikace v uvozovkách (libovolné jestli v jednoduchých nebo dvojitých).

Značky lze dále dělit na:

- Strukturní (nadpisy: `h1` až `h6`, odstavce: `p`)
- Popisné (`title`)
- Stylistické (`i`, `b`, `u`) – bývají stále častěji nahrazovány definicemi v kaskádových stylech

V současné době je připravována verze HTML 5. Příjem požadavků na oblast použitých značek jazyka byl ukončen 24.5.2011. Nová verze HTML je zaměřena především na zjednodušení zpracování ze strany prohlížečů, interakci s uživatelem, lepší propojení HTML, CSS 3 a JavaScript, zjednodušení specifikace vstupů, zlepšení práce s HW zdroji a širší podporu multimedií.

A.2 CSS

Cascading Style Sheets vznikl v roce 1997. CSS je jazyk zaměřující se na grafickou úpravu webových stránek. Odděluje informace a jejich strukturu na straně jedné od formátu jejich zobrazení na straně druhé. CSS je interpretováno prohlížečem při vypisování obsahu stránky a je tedy velmi závislé na aplikaci normovaných rysů ze strany prohlížeče. Pozitivní na CSS je skutečnost, že definice vizuální podoby je uvedena v jednom samostatném souboru a HTML soubory se na něj pouze odkazují. Změna v CSS má poté automaticky za následek i změnu grafické podoby ve vypisované stránce. Syntaxe kaskádových stylů je složena především z pravidel uvedených ve výpisu 7.

```
SelektorHTMLTagu {
    Vlastnost: hodnota;
```

```

}

.SelektorTridy {
    Vlastnost: hodnota;
}

#SelektorIdentifikatoru {
    Vlastnost: hodnota;
}

Selektor1, .Selektor2, #Selektor3 {
    Vlastnost: hodnota;
}
/* (hodnoty vlastností aplikovány na všechny selektory)*/

Selektor1 podSelektor2 .SelektorTridy3 {
    Vlastnost: hodnota;
}
/* (aplikováno jen na takto koncipovanou podstrukturu HTML kódu)*/

```

Výpis 7: CSS, ukázka syntaxe

Kaskádové styly je možno do stránek umístit třemi způsoby:

1. Samostatný CSS soubor a import v hlavičce HTML

```
<link rel="stylesheet" href="Styl.css" type="text/css">
```

Výpis 8: Tag pro import CSS souboru do hlavičky HTML

2. V hlavičce HTML

```

<style type="text/css">
<!--
    // definice CSS
-->
</style>

```

Výpis 9: Definice CSS v hlavičce HTML

3. V elementu v atributu style

```
<p style="margin-left: 15%;">Odstavec</p>
```

Výpis 10: Užití atributu style v HTML souboru

A.3 Microsoft SQL Server 2008

Microsoft SQL Server je relační databázový systém se zaměřením především na aplikace v oblasti obchodu nebo bankovníctví. První verze MS SQL Serveru vznikla na základě spolupráce Microsoft a Sybase (Sybase SQL Server) v roce 1992 (verze Microsoft SQL

Server 4.2). V roce 1995 se Sybase a Microsoft rozcházejí a Microsoft již sám vydává verzi 6.0.

Microsoft SQL Server nabízí řadu podpůrných nástrojů, jako například:

1. Profiler - pro zachycování událostí na serveru do logů a optimalizaci neefektivních procesů
2. Configuration Manager - pro správu služeb a síťového provozu
3. Database Engine Tuning Advisor – sada T-SQL funkcí pro optimalizaci indexů a pohledů bez nutnosti znát zevrubně strukturu databáze nebo technologii SQL Serveru
4. SQL CMD - pro zadávání SQL dotazů na příkazovém řádku
5. Replikační služby – pro provádění záloh a synchronizace více kopií

Microsoft SQL Server plně podporuje XML transakce. Pro získání dat z Microsoft SQL Serveru se používá (ANSI) SQL, případně T-SQL (specifické procedurální rozšíření SQL pro Microsoft SQL Server).

K dispozici je dále volně ke stažení SQL Server Management Studio (v případě instalace Visual Studio 2008 a vyšší je tento nástroj integrován přímo v instalačním balíčku). Tento nástroj nabízí přehledné GUI rozhraní pro správu databází.

SQL Server je možno pomocí modulů využít i v úlohách, jako jsou například datové sklady, dolování dat nebo business intelligence.

A.4 Internetová Informační Služba (IIS)

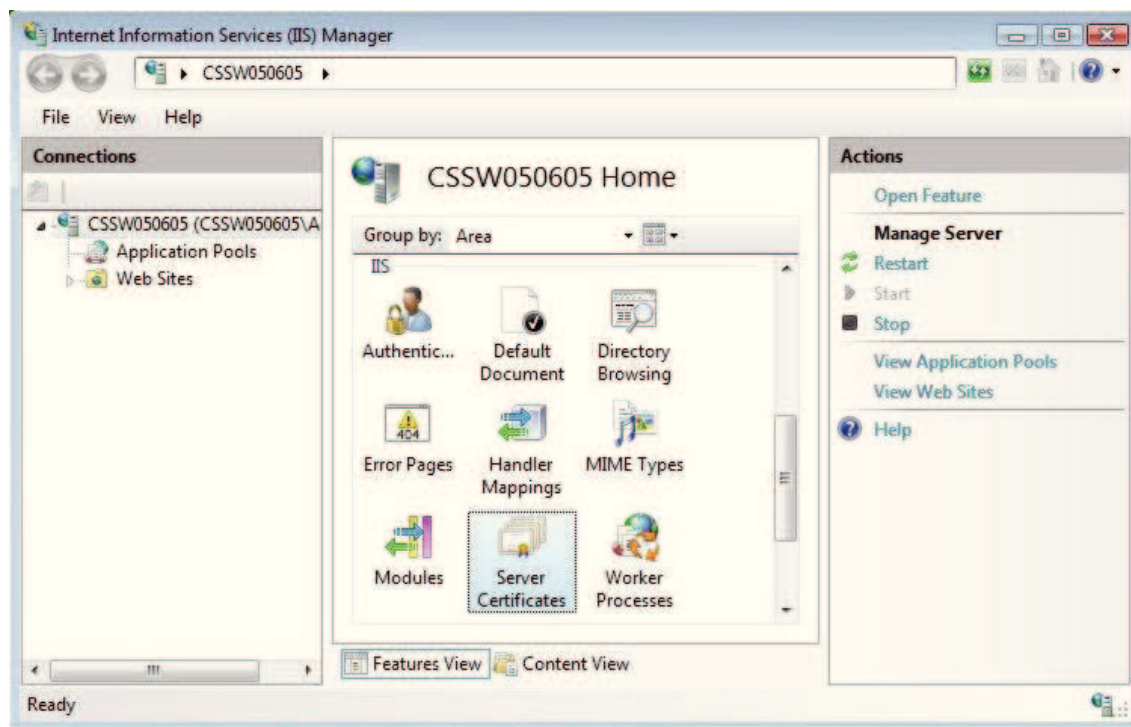
Internetová Informační Služba 7.5 (IIS, dříve Internet Information Server) je aplikace webového serveru vytvořená společností Microsoft pro použití s Microsoft Windows. Jedná se o 2. nepoužívanější webový server (nejvyžívanějším je Apache HTTP Server). IIS podporuje tyto síťové protokoly HTTP, HTTPS, FTP, FTPS, SMTP a NNTP. Je základní součástí řady systémů Windows Server, dále je také součástí některých edic systémů Windows XP, Vista a 7. IIS je také volně dostupná (jako Freeware) ve verzi IIS Express.

Dřívější verze IIS byly náchylné k řadě (někdy i vážných) chyb, především v oblasti bezpečnosti. V důsledku problémů s bezpečností byly v IIS 6.0 provedeny změny v chování předinstalovaných ISAPI, dále byla přidána funkce "Rozšíření webových služeb", která zabraňuje IIS spustit libovolný program bez vědomí správce systému.

V IIS 7 jsou komponenty ve formě modulů, takže je možné (a rozumné s ohledem na bezpečnost) instalovat jen to nejnutnější. Dále byla přidána funkce "Filtrování požadavků", která odmítá URL, jež administrátor nastavil jako podezřelé.

Instalovatelné moduly řeší celou řadu aspektů práce se serverem od administrace systému, či routování, přes proces nasazování webu, spravování multimedii až po management instalované databáze.

V červnu 2007, Google provedl studii 80 milionů domén a zjistil, že zatímco podíl IIS na trhu v té době byl 23%, hostoval 49% světového malware. To je stejně jako Apache



Obrázek 27: Administrace IIS

servery, jejichž podíl na trhu byl 66%. Studie také pozorovala zeměpisné polohy těchto "špinavých" serverů a došla k závěru, že příčinou by to mohlo být používání pirátských kopií Windows, pro které nebyly k dispozici aktualizace zabezpečení. To už neplatí. Microsoft dodává aktualizace zabezpečení pro všechny uživatele.

Zastoupení IIS v Internetu k 10.2011:

15,66 % serverů zpracovalo 12,46 % požadavků.

B Rysy používaných CMS systémů

Tato příloha obsahuje informace převzaté z porovnávacího katalogu:

<http://www.cmsmatrix.org/>. [15] Tyto informace bývají průběžně aktualizovány. Datum poslední aktualizace informací je uvedeno v záhlaví každé z tabulek, které vybrané CMS systémy popisují.

Veškeré zde uváděné systémy jsou **bezplatné** a **open-source**.

Nyní uvedu pro představu screenshoty následujících CMS systémů:

- Drupal
- PHP-Fusion
- PHP Nuke
- WordPress

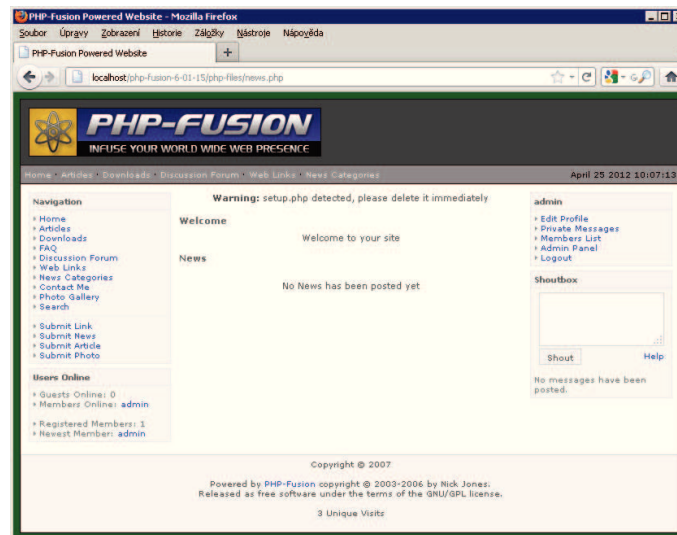
¹django-cms 1.2.1

²PHP-Fusion 6.00.206

³WordPress 3.0.4



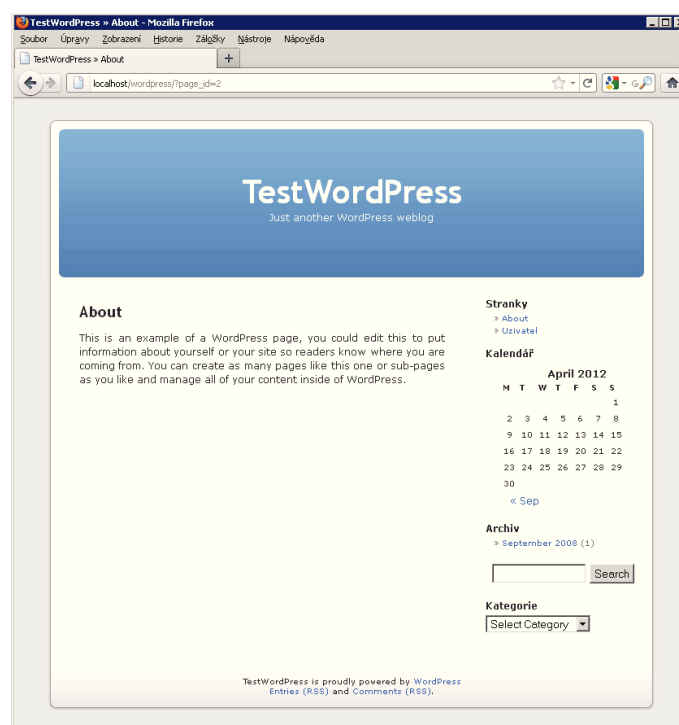
Obrázek 28: Administrace systému Drupal



Obrázek 29: Administrace systému PHP-Fusion



Obrázek 30: Hlavní strana systému PHP Nuke



Obrázek 31: Hlavní strana systému WordPress

Aktuální k	28.4.2011	2.8.2010	16.2.2012	15.2.2011	8.12.2007	2.7.2006	22.11.2005	14.4.2011
Systémové	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Webový Server			Apache	CGI	Apache	Apache		
DB Server	MySQL		MySQL	MySQL	MySQL	MySQL	MySQL	MySQL
OS	Nezávislé	Nezávislé	Nezávislé	Nezávislé	Nezávislé	Nezávislé	Nezávislé	Nezávislé
Programovací jazyk	PHP	Python	PHP	PHP	PHP	PHP	PHP	PHP
Bezpečnost	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Captcha	Ano	Modul	Modul	Modul	Ne	Ne	Ne	Ne
Schvalování obsahu	Omezeně	Ano	Ano	Ano	Ano	Ne	Ano	Ano
Kontrola emailem	Ano	Modul	Ano	Ano	Ano	Ne	Ano	Ano
Granularita oprávnění	Ano	Ano	Ano	Ano	Ano	Omezeně	Ano	Ano
Historie přihlášení	Ano	Modul	Ano	Ano	Ano	Ne	Modul	Modul
Authentikační moduly	Ano	Ano	Ano	Ano	Ano	Ne	Ne	Ano
Sandbox	Ano	Ne	Ne	Ne	Ano	Ne	Ne	Omezeně
Řízení sessions	Omezeně	Ne	Ano	Ano	Ano	Ne	Ne	Modul
SSL přihlašování	Omezeně	Ne	Modul	Ano	Ano	Ne	Ne	Ano
SSL stránky	Ano	Ne	Modul	Ano	Ano	Ne	Ne	Omezeně
Verzování	Ano	Modul	Ano	Modul	Ano	Ne	Ne	Modul
Podpora	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Manuály	Ano	Ano	Ano	Ano	Ano	Ne	Ne	Ano
Placená podpora	Ano	Ano	Ano	Ano	Ano	Ne	Ano	Ano
Školení	Ano	Omezeně	Ano	Ano	Ano	Ne	Ano	Ano
Online nápověda	Ano	Ano	Ano	Ano	Ano	Ne	Omezeně	Ano
Rozšiřitelné API	Ano	Ano	Ano	Ano	Ano	Ano	Ne	Ano
Profesionální hosting	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano
Profesionální služby	Ano	Ne	Ano	Ano	Ano	Ne	Ne	Ne
Diskusní fórum	Ano	Ne	Ano	Ano	Ano	Ano	Ano	Ano
Vývoj 3. strany	Ano	Ano	Ano	Ano	Ano	Ano	Ano	Ano

Tabulka 3: CMS - Systémové požadavky, Realizace bezpečnosti, Podpora

Aktuální k	28.4.2011	2.8.2010	16.2.2012	15.2.2011	8.12.2007	2.7.2006	22.11.2005	14.4.2011
Snadnost použití	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Drag and Drop správa	Ano	Ne	Modul	Ne	Ano	Ne	Ne	Ano
Hezké URL	Ano	Ano	Ano	Ano	Ano	Ne	Modul	Ano
Úpravy obrázků	Ano	Ano	Modul	Ano	Ano	Ne	Ne	Ano
Násobný upload	Ano	Ano	Modul	Ano	Ano	Ne	Ne	Ano
Šablony položek	Ano	Ne	Omezeně	Ano	Ne	Ne	Ne	Modul
Kontrola gramatiky	Ano	Ne	Modul	Modul	Modul	Ne	Ne	Ano
Odebírání informací	Ano	Ne	Modul	Ano	Ano	Ne	Modul	Modul
Šablonovací jazyk	Ne	Ano	Ano	Ano	Ne	Ne	Ne	Ne
Akce Zpět	Ano	Ano	Omezeně	Ne	Ano	Ne	Ne	Omezeně
WYSIWYG Editor	Ano	Ano	Modul	Ano	Ano	Ne	Ano	Ano
Výkon	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Cachování stránek	Ano	Ano	Ano	Ano	Ano	Ne	Ne	Modul
Zálohování databáze	Ne	Ano	Ano	Ne	Ano	Ne	Ano	Modul
Vyvažování výkonu	Ano	Modul	Ano	Ano	Ano	Ne	Ne	Ano
Export do statické verze	Ne	Ne	Ne	Ne	Ne	Ne	Ne	Modul
Řízení	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Reklamy	Modul	Modul	Modul	Ano	Ne	Ano	Ne	Ne
Časování zveřejnění	Ano	Ano	Modul	Ano	Ne	Ne	Ano	Omezeně
Hierarchie skupin	Ano	Ne	Ano	Ano	Ano	Ne	Ne	Ano
Styly	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano
Koš	Ano	Ne	Ne	Ano	Ne	Ne	Ne	Ano
Statistika webu	Modul	Modul	Ano	Ano	Ano	Ano	Ne	Modul
Rozhraní správa stylů	Ano	Ne	Ano	Ano	Ano	Omezeně	Ano	Ano
Rozhraní lokalizace	Ano	Ano	Ano	Modul	Ano	Ne	Ano	Omezeně

Tabulka 4: CMS - Snadnost použití, Výkon, Řízení

Aktuální k	28.4.2011	2.8.2010	16.2.2012	15.2.2011	8.12.2007	2.7.2006	22.11.2005	14.4.2011
Podpora formátů	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
RSS	Omezeně	Modul	Ano	Ano	Ano	Ne	Ano	Ano
FTP	Ano	Ne	Omezeně	Ano	Ano	Ne	Ano	Modul
UTF-8	Ano	Ano	Ano	Ano	Ano	Ne	Ano	Ano
XHTML	Ano	Omezeně	Ano	Ano	Ano	Ne	Ano	Ano
Flexibilita	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Uživatelská nastavení	Omezeně	Ne	Ano	Ano	Ano	Ne	Ne	Modul
Lokalizace rozhraní	Ano	Ano	Ano	Ano	Ano	Ano	Ne	Ano
Správa metadat	Omezeně	Ano	Ano	Ano	Ano	Ne	Ne	Ano
Vícejazyčný obsah	Ano	Ano	Ano	Modul	Ano	Ne	Ano	Modul
provázání	Ne	Ano	Ano	Modul	Ano	Ne	Ano	Modul
URL přesměrování	Ano	Ano	Ano	Ano	Ne	Ne	Ne	Ano

Tabulka 5: CMS - Podpora formátů, Flexibilita

Aktuální k	28.4.2011	2.8.2010	16.2.2012	15.2.2011	8.12.2007	2.7.2006	22.11.2005	14.4.2011
Moduly v instalaci	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Blog	Ano	Modul	Ano	Ano	Ano	Ano	Modul	Ano
Chat	Modul	Ne	Modul	Modul	Ano	Ne	Modul	Modul
Přístup podle práv	Ne	Ne	Modul	Modul	Ano	Ne	Ne	Modul
Evidence kontaktů	Omezeně	Ne	Modul	Ano	Ano	Ne	Omezeně	Modul
DB log	Modul	Ne	Modul	Modul	Ano	Ne	Omezeně	Ne
Diskusní fórum	Modul	Ne	Ano	Modul	Ano	Ano	Ano	Modul
Řízení dokumentů	Ne	Ne	Modul	Modul	Ano	Ne	Ano	Ne
Kalendář akcí	Ano	Ne	Modul	Modul	Ano	Ne	Modul	Modul
Často kladené dotazy	Ano	Ne	Ano	Ano	Ano	Ano	Ano	Modul
Distribuce souborů	Ano	Ne	Modul	Modul	Ano	Ano	Ano	Modul
Grafy a diagramy	Ne	Ne	Modul	Modul	Ano	Ne	Ne	Ne
Skupiny	Modul	Ne	Modul	Modul	Ano	Ne	Ne	Ne
Knihy hostů	Ano	Ne	Modul	Modul	Ano	Ne	Modul	Modul
Help Desk a hlášení chyb	Modul	Ne	Modul	Modul	Ano	Ne	Ano	Ne
In/Out Board	Ne	Ne	Modul	Ne	Ano	Ne	Ne	Ne
Úkoly	Modul	Ne	Modul	Modul	Ano	Ne	Ne	Modul
Správa odkazů	Modul	Ne	Modul	Ano	Ano	Ano	Ano	Ano
Formulář Odeslat email	Ano	Ne	Modul	Ano	Ano	Ne	Modul	Modul
Osobní stránka	Modul	Ne	Modul	Ne	Ano	Ne	Omezeně	Ano
Novinky	Ano	Ne	Modul	Modul	Ano	Ne	Modul	Modul
Fotogalerie	Ano	Modul	Modul	Modul	Ano	Ne	Ano	Ano
Ankety	Modul	Ne	Ano	Ano	Ano	Ano	Ano	Modul
Řízení produktů	Modul	Ne	Modul	Ano	Ano	Ne	Ano	Modul
Správa projektů	Ano	Ne	Modul	Modul	Ano	Ne	Omezeně	Ne
Vyhledávání	Ano	Modul	Ano	Ano	Ano	Ne	Ano	Ano
Mapa webu	Ano	Ano	Modul	Modul	Ne	Ne	Modul	Modul
Průzkumy	Modul	Ne	Modul	Modul	Ano	Ne	Ne	Modul

Tabulka 6: CMS - Moduly v instalačním balíčku

Aktuální k	28.4.2011	2.8.2010	16.2.2012	15.2.2011	8.12.2007	2.7.2006	22.11.2005	14.4.2011
Moduly v instalaci	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Průzkumy	Modul	Ne	Modul	Modul	Ano	Ne	Ne	Modul
RSS	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano
Testy, kvízy	Ne	Ne	Modul	Modul	Ano	Ne	Modul	Modul
Příspěvky uživatelů	Omezeně	Ne	Ano	Ano	Ano	Ano	Ano	Ano
Rozhraní webov. služeb	Modul	Ne	Omezeně	Ano	Ne	Ne	Ne	Modul
Komerce	Contao 2.9.4	django-cms ¹	Drupal 7.12	Joomla! 1.6.0	Moodle 1.9	PHP Nuke 6	PHP-Fusion ²	WordPress ³
Evidence průchodu webem	Modul	Ne	Modul	Modul	Ne	Ne	Ne	Ne
Nákupní košík	Modul	Ne	Modul	Modul	Ne	Ne	Ne	Ne
Předplatné	Ne	Ne	Modul	Modul	Ne	Ne	Ne	Ne
Poptávky	Modul	Ne	Modul	Modul	Ano	Ne	Ne	Ne

Tabulka 7: CMS - Moduly v instalačním balíčku 2

C Ostatní

C.1 Obsah přiloženého CD

Na přiloženém CD se nachází úplné zdrojové kódy a dokumentace k této práci. V následující tabulce je uvedena struktura přiloženého CD.

Adresář	Popis obsahu
/SRC/	Kompletní zdrojový kód v C#
/DOC/	Dokumentace zdrojového kódu
/Prilohy/	Další přílohy bakalářské práce
/Thesis/	Text bakalářské práce

C.2 Další přílohy

V dalších přílohách, které nejsou vytisknuty v této bakalářské práci (dokument v adresáři /Prilohy/) jsou uvedeny následující informace:

1. Rozbor případů užití Use-Case modelu
2. Rozbor datové vrstvy
3. Třídy v realizaci a jejich obecný popis
4. Zajímavé části realizace

C.3 Tiobe Index

V této příloze uvádím výpis Indexu oblíbenosti programovacích jazyků, který je připravován společností TIOBE Software [18]. Úplný název tohoto indexu zní: **TIOBE Programming Community Index**. Index je každé dva týdny v měsíci aktualizován a jsou do něj zařazeny nejrůznější jazyky pro nejrůznější platformy.

Uvedu zde jen tabulku 8 s dvaceti nejoblíbenějšími programovacími jazyky, TIOBE Index však analyzuje mnohem více informací. Celý Index je k dohledání na adrese: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.htm> [18].

Pozice a postup v žebříčku			Programovací jazyk	Zastoupení a nárůst [%]	
04.2012	04.2011	meziměsíčně		Zastoupení	Meziroční
1	2	+1	C	17.555	+1.39
2	1	-1	Java	17.026	-2.02
3	3		C++	8.896	-0.33
4	8	+4	Objective-C	8.236	+3.85
5	4	-1	C#	7.348	+0.16
6	5	-1	PHP	5.288	-1.30
7	7		(Visual) Basic	4.962	+0.28
8	6	-2	Python	3.665	-1.27
9	10	+1	JavaScript	2.879	+1.37
10	9	-1	Perl	2.387	+0.40
11	11		Ruby	1.510	+0.03
12	24	+10	PL/SQL	1.373	+0.92
13	13		Delphi/Object Pascal	1.370	+0.34
14	35	+10	Visual Basic .NET	0.978	+0.64
15	15		Lisp	0.951	+0.02
16	17	+1	Pascal	0.812	+0.10
17	16	-1	Ada	0.783	+0.01
18	18		Transact-SQL	0.760	+0.18
19	22	+3	Logo	0.652	+0.12
20	52	+10	NXT-G	0.578	+0.35

Tabulka 8: Nejoblíbenějších 20 programovacích jazyků (TIOBE Index)